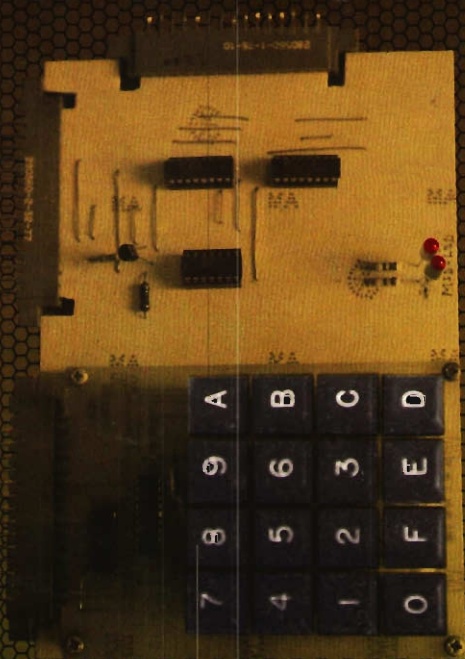
L. 2000

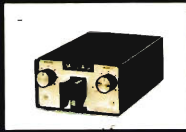
SUPPLEMENTO A cq ELETTRONICA N. 3

sped. in abb. post. g. III

• un byte da tastiera esadecimale • tutto sulle EPROM •
• bozza progetto VFO computerizzato • prova del nove •

G. LANZONI *è un fatto che dura nel tempo*

HAL — ROBOT — DOW KEY — TURNER — TEN-TEC DELTA

MILAG — HY GAIN — B. & W. — ROTORI - TRALICCI — POKET II/VHF

TEN-TEC — SAMSON — JUNKER — MILAG — DRAKE

BEARCAT — KENWOOD TR 7800 — KENWOOD TR 9000 — KENWOOD TR 2400 — KENWOOD TR 840

KENWOOD TS 770 — CAVI COASSIALI MIL. C. 17 E — VALVOLE

G. LANZONI

INOLTRE: ICOM - YAESU - AMPHENOL
Monitor (verdi) - Valvole
per tutti gli apparati di oggi
e di ieri ... e migliaia di ricambi

TEL. (02) 589075 - 544744

20135 MILANO - VIA COMELICO, 10

SUPPLEMENTO cq ELETTRONICA

sommario

4 una scheda video per il vostro μp (Vidmar)

20 Interfacciamo la TI-57 (Ibridi)

27 Tutto quello che avreste voluto sapere sulle EPROM... (Sinigaglia)
...(e che non avete mai osato chiedere)

36 Bozza di progetto per un VFO computerizzato (Becattini)

44 GP User's Group

46 un byte da una tastiera esadecimale (Prizzi)

56 Modulo pilotabile direttamente dal Data Bus del μp (Casaroli)

60 Acquisizione dati da otto canali analogici per microcomputer (Anselmi)

72 La prova del nove (Crispa)

indice degli inserzionisti di questo numero

nominativo	pagina	nominativo	pagina	nominativo	pagina
BARLETTA App. Scient.	**4. di copert.**	**FIRENZE 2**	**55**	**LANZONI**	**2. di copertina**
EDIZIONI CD	**59-81**	**FONTANA elettronica**	**58**	**MARCUCCI**	**82**
FALCON	**26**	**GRIFO**	**43**	**SELCOM elettronica**	**3. di copertina**

EDITORE s.n.c. edizioni CD
DIRETTORE RESPONSABILE Giorgio Totti
REDAZIONE - AMMINISTRAZIONE
ABBONAMENTI - PUBBLICITÀ
40121 Bologna-via C. Boldrini, 22-(051) 552706-551202
Registrazione Tribunale di Bologna, n. 3330 del 4-3-1968

STAMPA: Tipo-Lito Lame - Bologna - via Zanardi, 506/B
Spedizione in abbonamento postale - gruppo III
Pubblicità inferiore al 70%
DISTRIBUZIONE PER L'ITALIA
SODIP - 20125 Milano - via Zuretti, 25 - ☎ 6967

DISTRIBUZIONE PER L'ESTERO
Messaggerie Internazionali - via Gonzaga, 4 - Milano
Cambio indirizzo L. 1.000 in francobolli
Manoscritti, disegni, fotografie,
anche se non pubblicati, non si restituiscono

ABBONAMENTO Italia a 12 mesi L. 24.000 (nuovi)
L. 23.000 (rinnovi)
ARRETRATI L. 2.000 cadauno
Raccoglitori per annate L. 7.500 (abbonati L. 7.000).

TUTTI I PREZZI INDICATI comprendono tutte le voci di spesa (imballi, spedizioni, ecc.) quindi null'altro è dovuto all'Editore.

SI PUÒ PAGARE inviando assegni personali e circolari, vaglia postali, o a mezzo conto corrente postale 343400, o versare gli importi direttamente presso la nostra Sede. Per piccoli importi si possono inviare anche francobolli da L. 100.

A TUTTI gli abbonati, nuovi e rinnovi, sconto del 10% su tutti i volumi delle edizioni CD.

ABBONAMENTI ESTERO L. 27.000
Mandat de Poste International
Postanweisung für das Ausland
payable à / *zahlbar an*

edizioni CD
40121 Bologna
via Boldrini, 22
Italia

una
scheda video
per il vostro
microprocessore

YU3UMV, ing. Matjaž Vidmar

Introduzione

Su varie riviste di elettronica sono stati già pubblicati diversi schemi di terminali video e schede d'interfaccia video per microcomputer.
Anche sul mercato sono disponibili diverse soluzioni: apparecchi completi, montati e inscatolati, schede premontate e apparecchi in forma di kit.
Analizzando però le varie soluzioni circuitali adottate si scoprono parecchi difetti:

- **La flessibilità d'impiego:** *le funzioni dell'interfaccia video sono generalmente realizzate o con un integrato LSI dedicato, un "CRT controller", oppure con alcune dozzine d'integrati TTL; in ogni caso le funzioni sono realizzate via hardware, perciò è molto difficile fare delle modifiche e/o aggiungere funzioni nuove.*
- **La reperibilità dei componenti:** *i generatori di caratteri (memorie ROM) e i CRT controllers non rappresentano nulla di nuovo per quanto riguarda la tecnologia di costruzione se confrontati con i vari microprocessori, RAM e EPROM reperibili in commercio; mentre per le RAM e le EPROM i tipi sono standardizzati e tutti i produttori di circuiti integrati producono gli stessi tipi di memorie, i generatori di caratteri e ancora più i CRT controllers non sono standardizzati e ogni fabbrica produce i suoi integrati che, anche se simili agli integrati della concorrenza, non sono quasi mai equivalenti.*
- **Il costo dell'apparecchio:** *sembra che i progettisti di questi apparecchi non si preoccupino del costo della realizzazione: generatori di caratteri, CRT controllers costosissimi e costose PROM difficili da reperire prima e da programmare poi.*

Progettando la scheda video che presento in questo articolo ho cercato di ovviare a questi inconvenienti: naturalmente era necessario un compromesso tra le varie richieste.
Innanzitutto ho deciso di collegare la scheda video direttamente al bus del microprocessore. La scrittura nella memoria RAM della scheda video viene eseguita dal microprocessore. In questo modo tutte le funzioni di scrittura vengono eseguite dal software: scrittura carattere, nuova linea, ritorno all'inizio della linea, cancellazione riga e pagina, scroll e cursore. Per modificare le funzioni o aggiungerne delle nuove basta modificare il programma.
L'organizzazione della RAM della scheda video è di 4k x 8 bit (vedi figura 1).

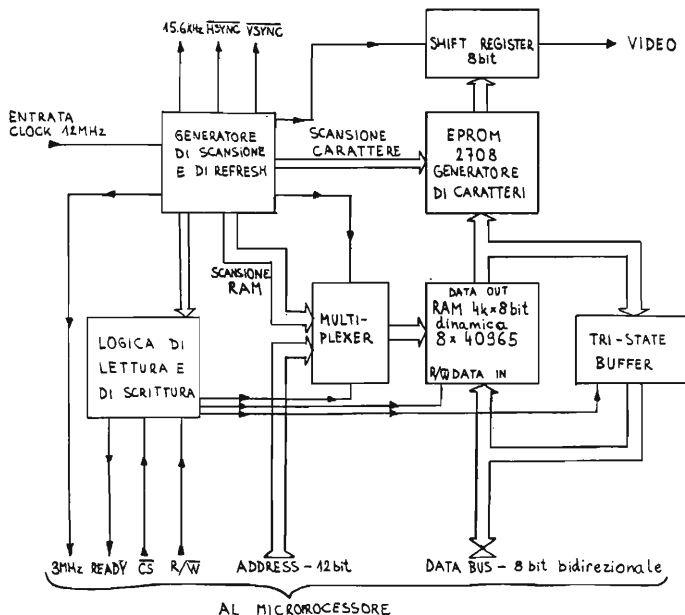



figura 1
Schema a blocchi della scheda video.

Metà della RAM è dedicata alla rappresentazione di 32 righe di 64 caratteri ciascuna mentre l'altra metà può essere liberamente impiegata dal microprocessore per non rallentare troppo il funzionamento di quest'ultimo.
In commercio sono reperibili vari tipi di generatori di caratteri, personalmente ho però preferito programmare una EPROM con i seguenti vantaggi:
- i generatori di caratteri commerciali hanno generalmente 64 caratteri ASCII (lettere maiuscole e cifre), programmando una EPROM 2708 si possono invece ottenere 128 caratteri: tutti i 96 caratteri ASCII (lettere maiuscole, minuscole e cifre) e in più 32 caratteri grafici;

- la EPROM 2708 è un NMOS e ha un tempo d'accesso circa due volte inferiore dei generatori di caratteri commerciali, che sono dei PMOS; questo fatto semplifica notevolmente il progetto della scheda video (non sono necessari dei latch TTL);

- la 2708 è facilmente reperibile e costa meno dei generatori di caratteri commerciali.

La logica attorno alle memorie deve essere assai veloce, la frequenza di clock è di 12 MHz. Per non avere problemi di temporizzazione ho impiegato i comuni TTL, comunque nulla vieta di sostituirli con la serie LS a consumo ridotto.

Descrizione del circuito

Tutte le frequenze necessarie al funzionamento della scheda vengono ottenute tramite la divisione del clock a 12 MHz (vedi figura 2).
I due 74161 e il 74112 dividono i 12 MHz per 768 (16x16x3); alle uscite del 74112 troviamo la frequenza di linea TV di 15.625 Hz. Metà del 74109, il 74193 e il 7490 dividono questa frequenza per 320 (2x16x10) per ottenere la frequenza di quadro. Il quadro ha perciò 320 linee al posto delle 312,5 linee dello standard CCIR, i TV-monitor non sono molto sensibili alle piccole variazioni della frequenza di quadro. Tramite opportuni circuiti decodifica si astraggono i segnali di sincronismo HSYNC e $\overline{\text{VSYNC}}$. Ho preferito eseguire le uscite dei segnali di sincronismo e l'uscita del segnale video separate, in questo modo si possono pilotare anche direttamente i circuiti di deflessione del CRT e lo stadio finale video. Per pilotare un TV monitor si può aggiungere un mixer (vedi più avanti alla figura 8).
Una linea TV dura 64 μs, considerando il tempo di ritraccia e un certo margine ai bordi dell'immagine rimangono 40 ÷ 45 μs utili. Con il circuito presentato un carattere dura 666 ns (frequenza di ripetizione 1,5 MHz) e una riga di 64 caratteri occupa 42,6 μs ($\frac{2}{3}$ dell'intera riga TV). Il primo 74161 con le relative porte di decodifica ha il compito di generare gli impulsi a 1,5 MHz per pilotare la lettura dalle RAM: impulsi $\overline{\text{RAS}}$, $\overline{\text{CAS}}$ e trasferimento nello shift register $\overline{\text{PE}}$.
I caratteri sono formati da un mosaico di 8 punti x 8 linee; i caratteri alfanumerici sono formati da 7 punti x 7 linee utili, una linea e una colonna vengono utilizzate per la separazione tra i caratteri. I dati di una linea di un carattere (8 bit) vengono letti in parallelo dal generatore di caratteri e caricati in un veloce shift register TTL, all'uscita serie dello shift register si ottiene il segnale video. La frequenza di clock dello shift register è 12 MHz, cioè otto volte la frequenza di ripetizione dei caratteri. La larghezza di banda del segnale video ottenuto è perciò 6 MHz minimi.
Una riga ha 64 caratteri, per indirizzarli è necessario fornire 6 bit di address ($2^6 = 64$) alla memoria RAM (HAO ÷ HA5 su figura 2).
I caratteri sono alti 8 linee, 32 righe di caratteri occupano perciò 256 linee TV. Dalle 320 linee del quadro rimangono 64 linee per la ritraccia verticale e un certo margine in cima e in fondo dell'immagine. Per scansionare le 32 righe del testo sono necessari 5 address verticali VA0 ÷ VA4. Durante la scansione di una riga del testo (8 linee TV) gli address verticali della RAM (VA0 ÷ VA4) rimangono fissi, viene invece scansionato il generatore di caratteri tramite RS0, RS1 e RS2.
Le otto RAM impiegate sono da 4k x 1 bit dinamiche con gli address multiplexati (vedi figura 4).

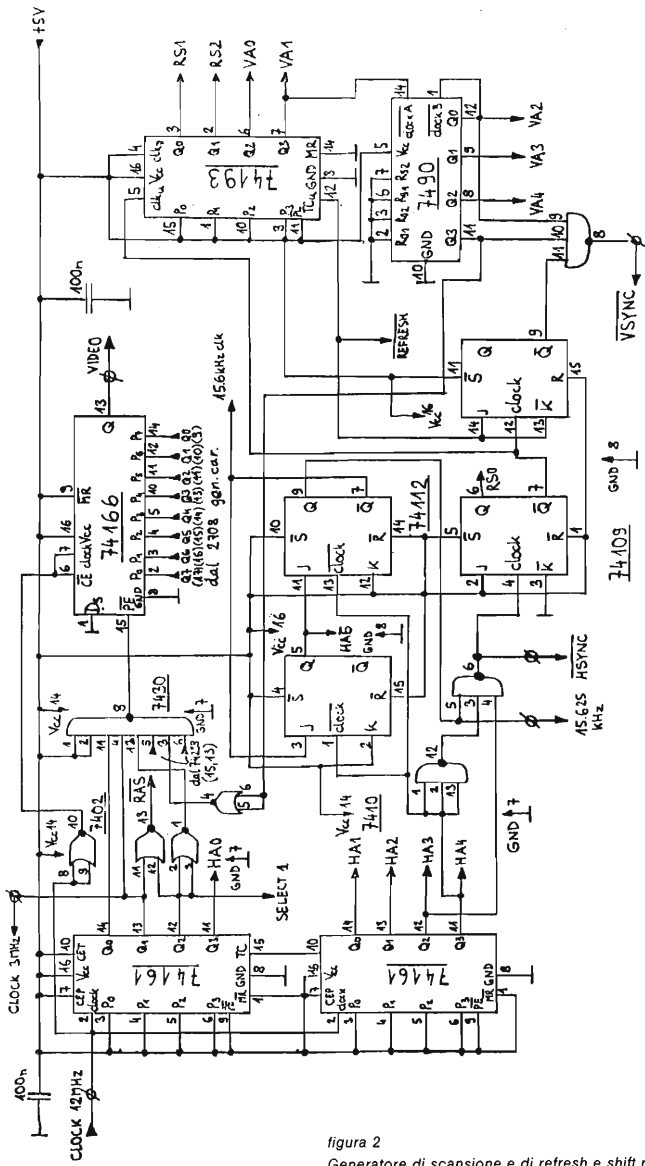
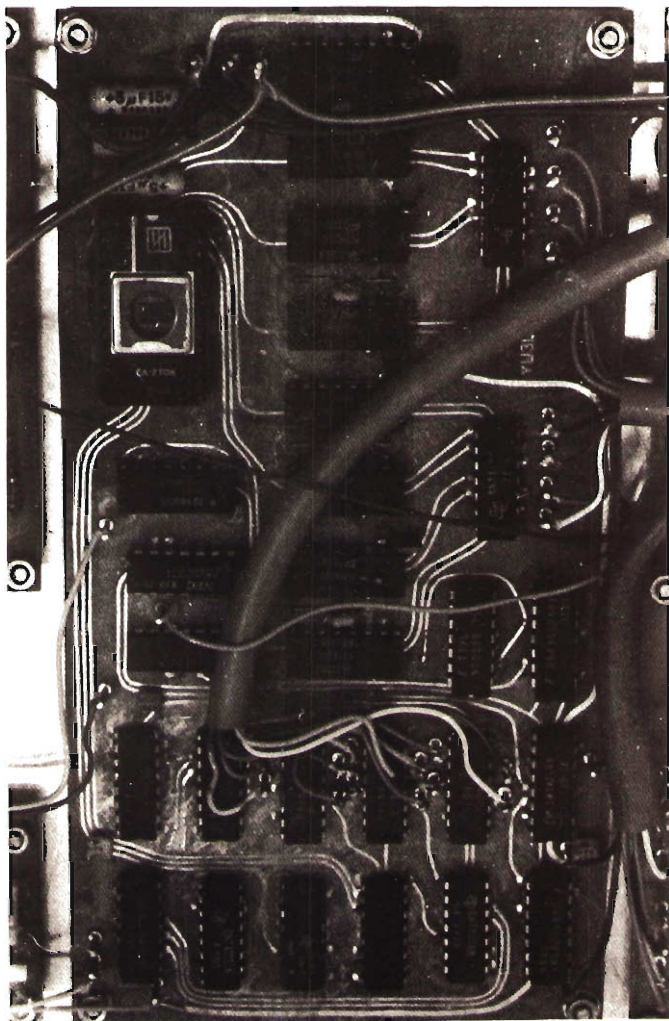



figura 2
Generatore di scansione e di refresh e shift register.

Il multiplexer costituito dai tre 74153 (vedi figura 3) ha due scopi:
- di multiplexare i 12 bit necessari per indirizzare 4k di memoria sulle 6 linee di address delle memorie MA0 ÷ MA5;
- di commutare gli address della memoria tra il microprocessore e la logica di scansione della scheda video.

La scheda video collegata a un microcomputer.

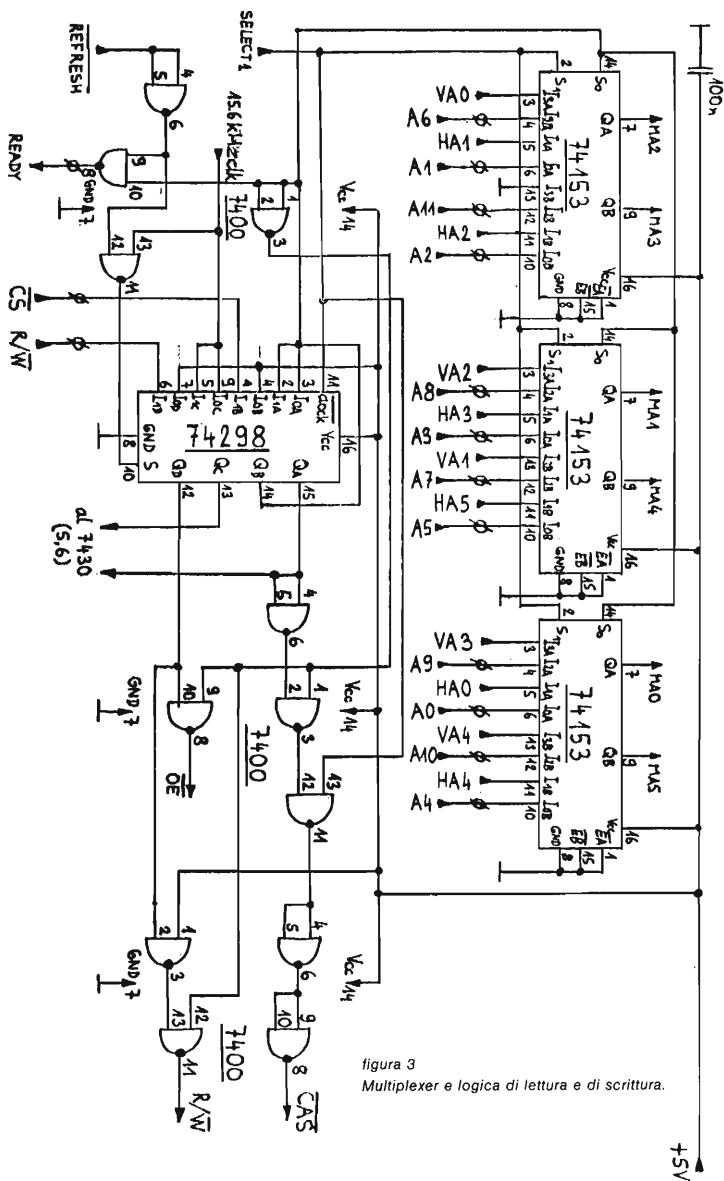



figura 3
Multiplexer e logica di lettura e di scrittura.

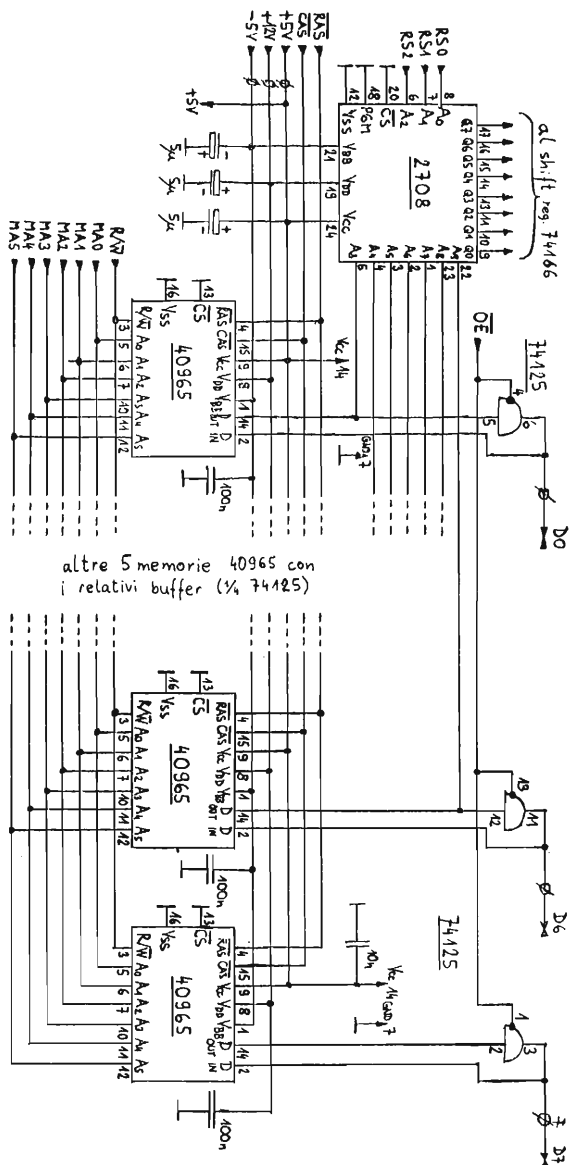



figura 4
RAM dinamica e generatore di caratteri.

Il microprocessore chiede di comunicare con la memoria RAM (leggere o scrivere) inviando un livello logico basso all'entrata $\overline{CS}$ della scheda video. Poiché i circuiti della scheda video generalmente non sono sincronizzati con il microprocessore, per primo devono completare il presente ciclo di lettura. La sincronizzazione viene effettuata dal latch 74298. Le memorie ricevono sempre gli stessi clock $\overline{RAS}$ e $\overline{CAS}$, però tramite il multiplexer ricevono gli address dal microprocessore. Nel caso di una lettura della RAM vengono anche abilitati i tri-state buffer che connettono le uscite delle memorie al data bus del µp.
Quando le memorie RAM comunicano col µp non possono allo stesso tempo inviare i dati giusti al generatore di caratteri. Il disturbo si presenta come uno sfarfallio sullo schermo TV. Per minimizzare l'effetto del disturbo la logica blocca nel momento opportuno il trasferimento dei dati dal generatore di caratteri nello shift register. In questo modo la parte mancante del carattere viene sostituita da un trattino scuro.
Per la sincronizzazione della richiesta del microprocessore è richiesto un certo tempo, perciò il tempo complessivo d'accesso alla memoria può arrivare anche a 1,3 µs. La soluzione più semplice è di ridurre la frequenza del clock del µp per adattarlo alla memoria della scheda video.
Le memorie RAM dinamiche richiedono per il mantenimento dei dati dei cicli di refresh, generalmente ogni 2 ms. Le RAM dinamiche impiegate richiedono 64 cicli di lettura o scrittura sui 64 possibili row address ogni 2 ms per il refresh. La memoria RAM è collegata al generatore di scansione in modo che durante una linea TV vengono richiamati tutti i 64 row address. Il problema del refresh si presenta perciò soltanto quando il microprocessore chiede di comunicare con la memoria RAM. Per garantire il refresh su tutti i 64 row address anche nel caso peggiore, ogni 32 linee TV (32 x 64 µs = 2.048 ms) la logica viene forzata nella scansione normale (linea $\overline{REFRESH}$).Il microprocessore viene preavvisato tramite la linea READY che la memoria RAM non è disponibile.

Costruzione

I componenti della scheda video sono montati su un circuito stampato a doppio rame (vedi figure 5, 6 e 7).
Purtroppo sugli schemi delle figure 2, 3 e 4 non era possibile disegnare i condensatori di disaccoppiamento delle linee di alimentazione nelle loro esatte posizioni sul circuito stampato. Buona parte delle transizioni dal lato componenti al lato rame è realizzata dai piedini dei circuiti integrati, saldati da entrambi i lati, poiché non avevo la possibilità di metallizzare i fori della piastrina. Le rimanenti transizioni devono invece essere realizzate da spezzoni di filo saldati da entrambi i lati. Tutti gli integrati MOS impiegati hanno gli ingressi protetti da almeno un diodo tra l'ingresso e il substrato V_{BB}. Si può facilmente controllare la presenza di questo diodo con un ohmetro, portata Ω x 100 o Ω x 1.000, senza danneggiare l'integrato. Gli integrati MOS impiegati possono perciò essere saldati come i comuni TTL. Consiglio di montare su zoccolo soltanto la EPROM 2708 in modo da poter facilmente cambiare il character font. Gli integrati NMOS impiegati richiedono tre tensioni d'alimentazione: –5 V, +5 V e +12 V. I +5 V servono soltanto per interfacciare i TTL, gran parte del circuito interno dei NMOS viene alimentata dai +12 V. I –5 V connessi al substrato V_{BB} hanno circa la stessa funzione che aveva la tensione negativa sulla griglia controllo delle valvole. Se per un guasto all'alimentatore oppure per una interruzione della pista dei –5 V o un corto viene a mancare la tensione negativa sul piedino V_{BB}, gli integrati NMOS possono essere distrutti a causa della sovradissipazione.

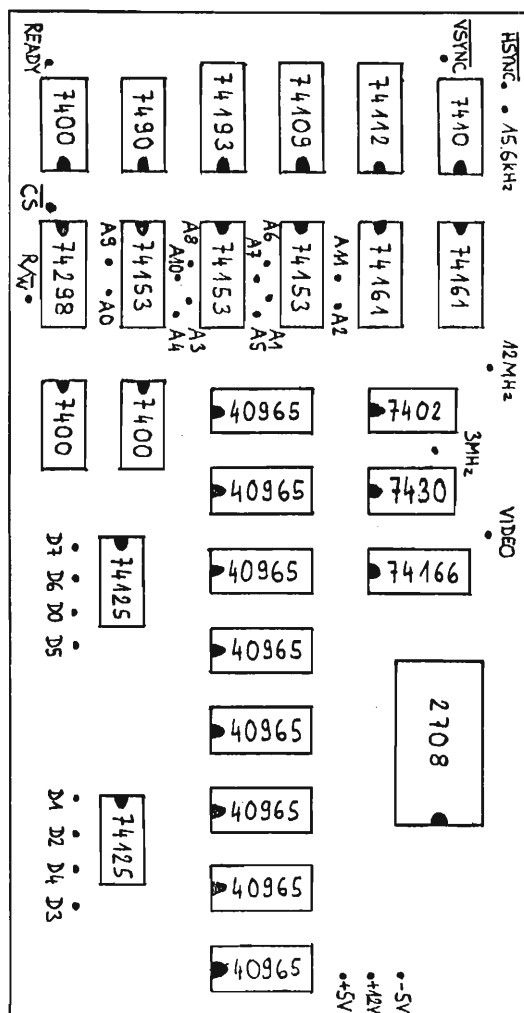



figura 5

Disposizione dei componenti principali e i collegamenti (vista lato componenti).

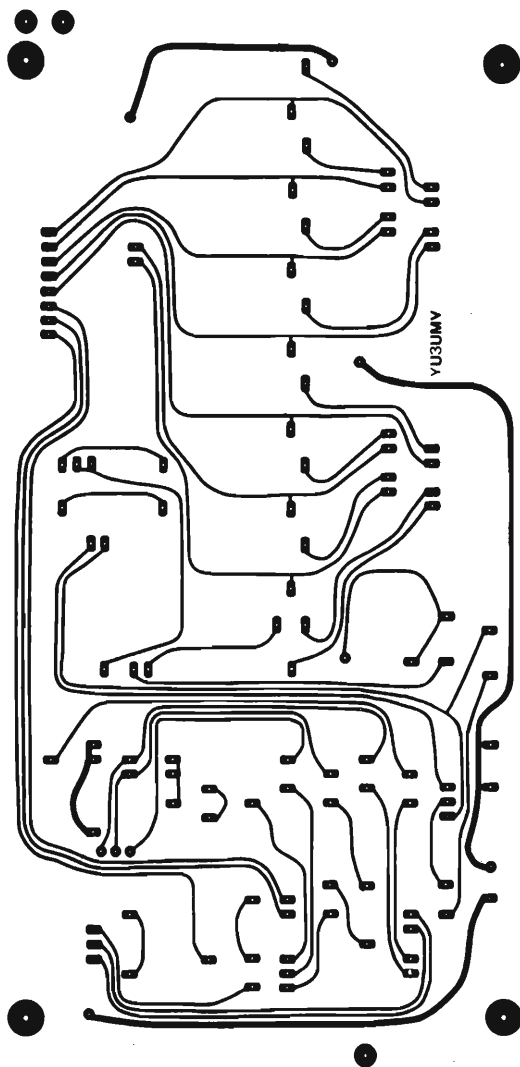



figura 6
Circuito stampato, doppio rame - lato componenti.

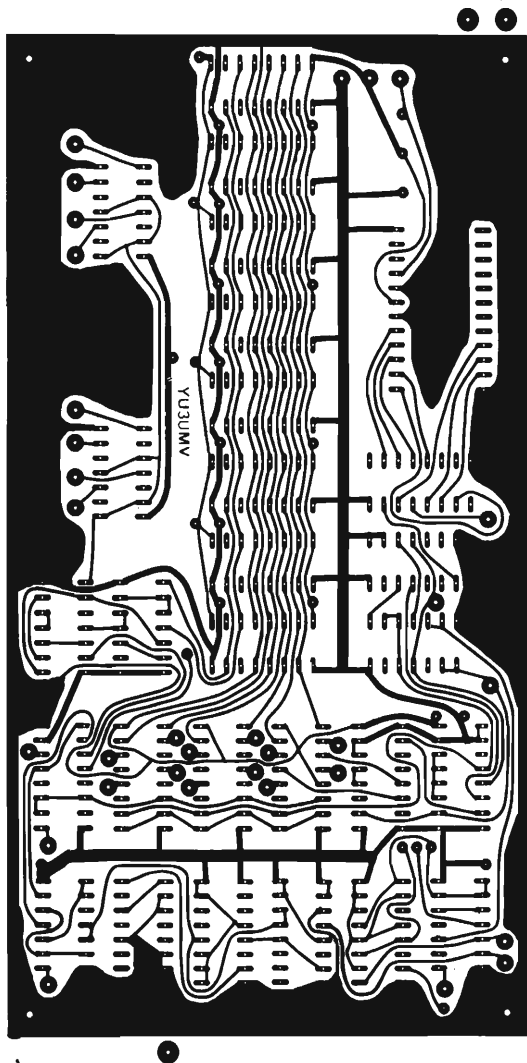



figura 7
Circuito stampato, doppio rame - lato rame.

In commercio sono reperibili vari tipi di memorie RAM dinamiche da 4k bit. Sullo schema sono disegnate le 40965 della Fairchild, gli equivalenti diretti sono la MK 4096 della Mostek, la 2104 della Intel, la 2660 della Signetics e la 6604 della Motorola. Possono però essere impiegati anche altri tipi di memorie dinamiche da 4k bit con gli address multiplexati in custodia a 16 pin, a patto che abbiano il latch all'uscita D_{OUT}. Le memorie vengono selezionate per il tempo d'accesso dalla Casa produttrice. In questo circuito il timing non è critico e possono essere impiegate anche le memorie più lente (350 ns RAS access time) e quindi più economiche. Anche per la EPROM generatore di caratteri va bene il tipo standard con 450 ns di tempo d'accesso.
Nella realizzazione ho impiegato gran parte integrati di recupero dalle schede surplus. Anche impiegando integrati nuovi sconsiglio però la costruzione "alla cieca". Molte volte succede che anche gli integrati nuovi presentano dei difetti. Le Case costruttrici garantiscono generalmente l'1% o meno esemplari difettosi per la merce di prima scelta, l'esperienza però insegna che non è detto che gli integrati pagati cari siano in realtà di prima scelta. Per poter portare a termine la realizzazione è essenziale conoscere il principio di funzionamento del circuito e disporre di un minimo di strumentazione (oscilloscopio).

Collegamento della scheda video

La scheda video richiede per il funzionamento un clock esterno a livello TTL a 12MHz. Una soluzione economica è rappresentata in figura 8.

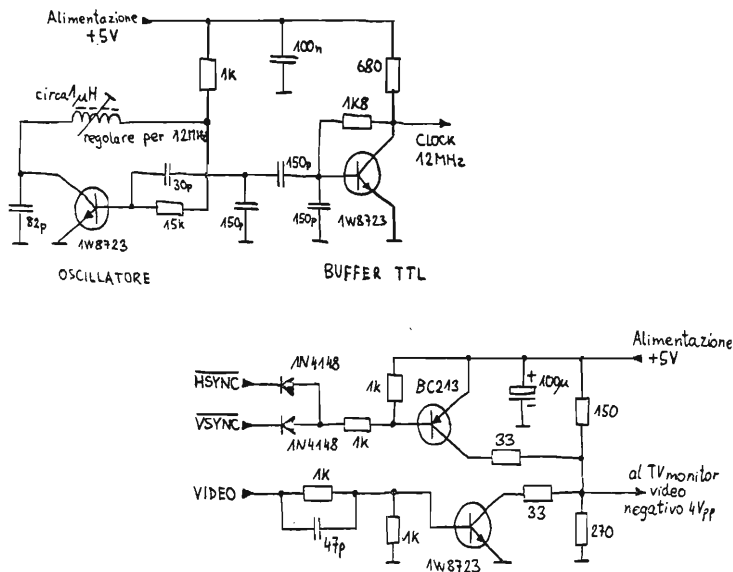
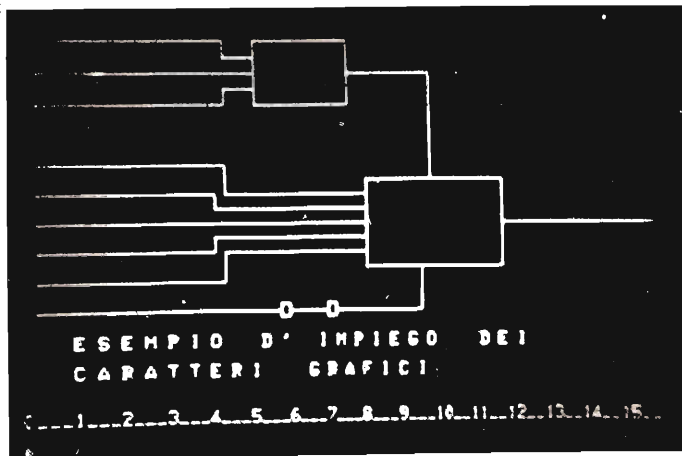



figura 8
Circuiti suggeriti per completare la scheda video.

Per pilotare un TV monitor è necessario combinare i segnali di sincronismo con il segnale video utile. Con il circuito in figura 8 i caratteri appaiono bianchi su fondo scuro.
Il data bus della scheda video può essere connesso direttamente al data bus del microprocessore. Grazie ai buffer "tri-state", il data bus della scheda video rappresenta soltanto un carico capacitivo (tranne che in fase di lettura), perciò non carica il data bus del µp e non provoca problemi di pilotaggio da parte del microprocessore. Gli ingressi del multiplexer della scheda video sono degli ingressi TTL e caricano l'address bus del µp. L'ingresso $\overline{CS}$ della scheda video va collegato al decoder degli address più significativi.
Personalmente ho impiegato la scheda video assieme al microprocessore 8080A. Per interfacciare senza problemi la scheda video ho ridotto la frequenza del clock dello 8080A a 750 kHz (la frequenza massima per lo 8080A è di 2 MHz). Lo 8080A viene generalmente impiegato assieme al 8224 generatore di clock e al 8228 data bus buffer e interrupt. L'unico difetto di questi due ultimi integrati è di costare assieme quasi il doppio dello 8080A. Le loro funzioni possono però essere svolte da pochi TTL di recupero delle schede (vedi figura 9).
Un 74112 divide per 4 la frequenza di 3 MHz e assieme al 7403 genera i due clock $Ø_1$ e $Ø_2$ per il microprocessore. Il 74298 sincronizza i segnali di reset e di ready e memorizza il bit D0 dello stato dello 8080A (segnale INTA). La logica con il 7400 pilota le memorie (segnale ENABLE) e il circuito per la generazione dell'istruzione RST 7 nel caso di INTA, che in esadecimale corrisponde a FF.
Le subroutines di gestione della scheda video come terminale ASCII occupavano circa 300 bytes di memoria.

Rappresentazione su monitor TV da 6". Consiglio però l'impiego di un monitor da almeno 9".

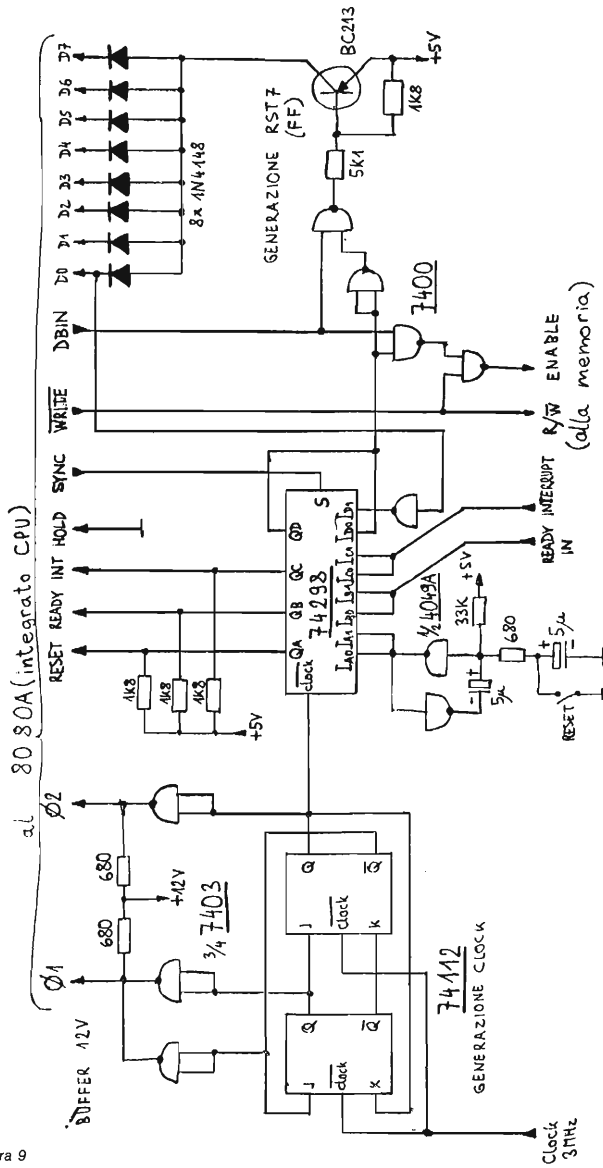
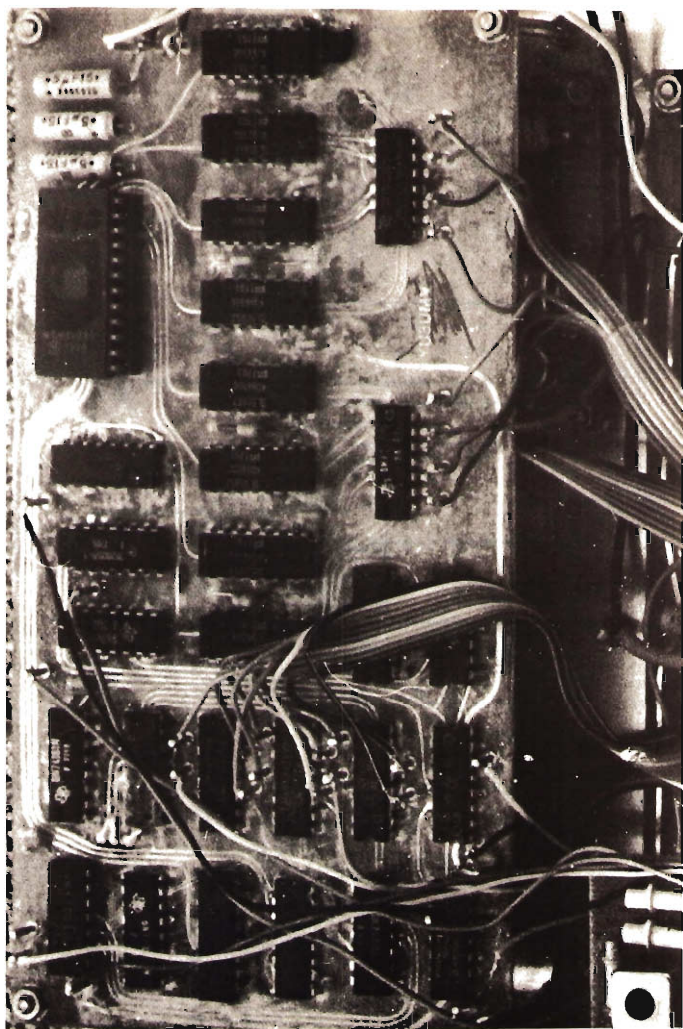
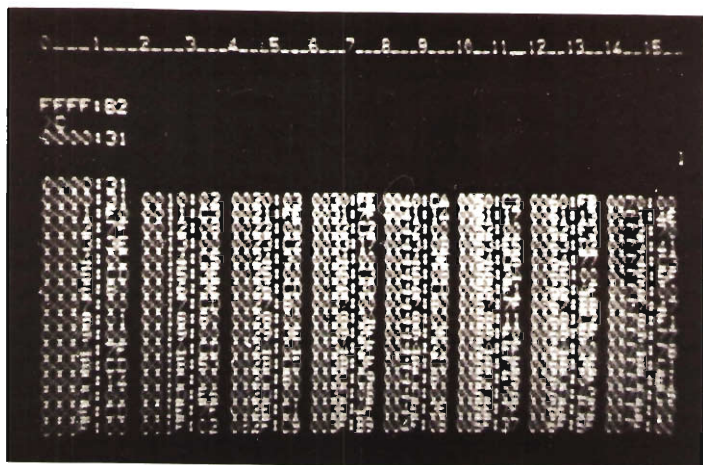



figura 9
Un'idea-spunto per la generazione del clock per lo 8080 e la generazione dell'interrupt.

Un'altra scheda video collegata a un altro microcomputer.

Listing di 128 bytes di programma.

Conclusioni

La scheda video che ho presentato in questo articolo non rappresenta l'ultimo grido della tecnologia e le soluzioni circuitali adottate sulla scheda non permettono al microprocessore di lavorare alla sua massima velocità. Come spiegato nell'introduzione, ho cercato di ottenere un **compromesso** tra le varie richieste.
Riprogettando la scheda si potrebbero adottare varie modifiche: impiegare una EPROM 2716 per avere un character font di 256 caratteri, sostituire le obsolete RAM dinamiche da 4k bit con RAM dinamiche da 16k bit oppure RAM statiche da 4k bit.
Forse il difetto maggiore della scheda video è di non essere un apparecchio completo e indipendente, visto che è predisposta per essere collegata direttamente al bus di un microprocessore. Naturalmente è possibile realizzare su di una scheda separata la logica di scrittura, con integrati TTL o CMOS, però oggigiorno la soluzione a microprocessore è decisamente più economica.
Rimango a disposizione per eventuali chiarimenti e/ o duplicazione EPROM. ❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊❊

Interfacciamo la TI-57

I4IBR, Marco Ibridi

"Interfacciare" è certamente uno dei verbi più in voga in questo periodo in cui l'informatica galoppa verso mete pressoché inimmaginabili, vuoi a cavallo di magnifici puledri IBM o altri, vuoi in sella a splendidi Apple, Pet o Trs-80 ma anche sulle tastiere di straordinarie programmabili dalle possibilità fantascientifiche solo fino a qualche anno fa.
Ciò che io voglio ora proporre è la possibilità di dare, per l'appunto, una voce nuova a una magnifica programmabile quale è la TI57 della Texas Instruments.

Detta programmabile è infatti il cavallo di battaglia di parecchie persone che, piu o meno appassionate di elettronica "vulgaris", vogliono scoprire cosa si nasconde dietro questo appassionante mondo dell'"intelligenza artificiale" e che allo stesso tempo non vogliono impegnare cifre considerevoli.
Il prezzo infatti è ciò che strizza più l'occhiolino ma non ci dobbiamo fermare qui, la '57 è un vero, seppur piccolo, computer.
Presto però tutti, chi prima chi poi, ci si accorge con rammarico dei limiti che detta macchina ci impone e allora cominciano i pensieri.
C'è chi pensa subito di passare alle sorelle maggiori e chi invece, come me, decide di guardarci un po' "dentro".
Appaiono così, un po' su tutte le riviste del ramo, vari suggerimenti e varie modifiche, ma quasi tutte si limitano a usarla né piu né meno come timer intelligente. Certamente io non ho scoperto l'America, chissà a quanti prima di me sarà venuta l'idea, ma ho tentato di "nobilitarla" dandole la possibilità di colloquiare, o meglio "ascoltare" il mondo esterno.

Di darle un 'bus' più o meno standard nemmeno a pensarlo e allora la classica lampadina: perche non interfacciare la tastiera? Da qui ogni sorta di immondo pensiero, dalla possibilità di registrare i programmi su nastro magnetico, per i più bravi e i più zelanti, o su banda perforata, per i più pigri come me, e via sulla strada dell'implementazione di giochi o cose più utili. Quello che vi presenterò, più che un progetto finito e a sé stante è una **strada nuova** su cui sbizzarrirvi con la vostra splendida '57.

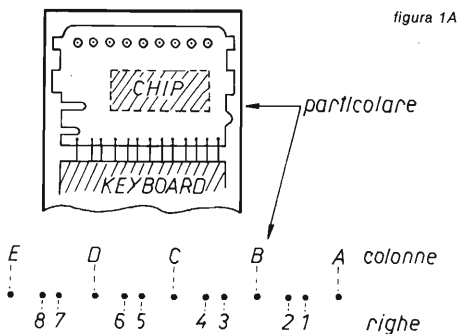
I contatti della tastiera sono perfettamente visibili nello spaccato di figura 1A e sono "quantificabili" nelle lettere da me assegnate, in modo assolutamente arbitrario, in calce a detta figura.

facile per tutti

figura 1A

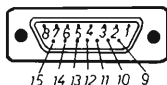
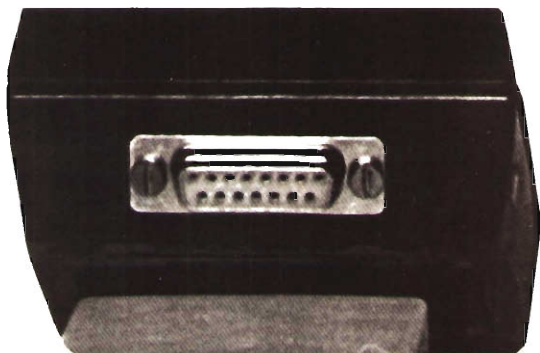



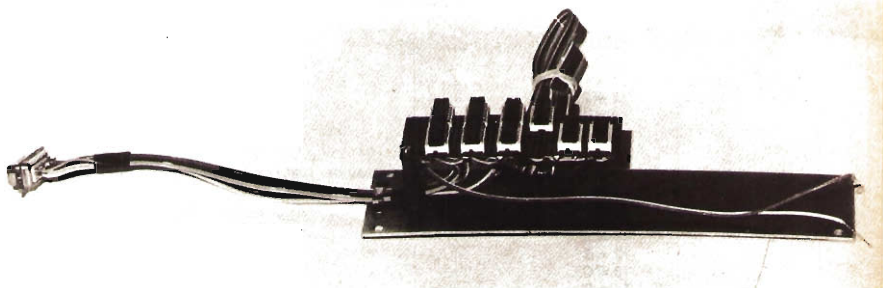
figura 1B

1 = 1
2 = 2
3 = 3
4 = 4
5 = 5
6 = 6
7 = 7
8 = 8
9 = n.c.
10 = n.c.
11 = A
12 = B
13 = C
14 = D
15 = E

figura 1C

Collegamenti consigliati per l'utilizzo del connettore "Cannon"

La relazione che lega detta matrice con i simboli risultanti è data dalla figura 1B, che si rifà per l'appunto al tastierino della macchina in questione; il tutto elegantemente realizzato con micro connettori Cannon, acquistabili alla GBC e abbastanza "salati", a 15 poli come da figura 1C e relative foto.

A questo punto la domanda: cosa ne facciamo? Ai più bravi nessuna risposta, o hanno gia snobbato tutto quanto o hanno già deciso cosa fare, agli altri, forse non meno bravi ma più curiosi, illustrerò parte della mia mirabolante costruzione.

L'idea base è quella di far corrispondere a ogni tasto un ben determinato byte di sette bit, otto nel caso si inserisca un particolare bit di controllo, in modo da poter memorizzare o manipolare il dato con relativa facilità.
Infatti, se osserviamo di nuovo la figura 1B notiamo che, per ottenere, ad esempio, il carattere '8' si deve congiungere la riga n. 7 con la colonna 'C', come pure per ottenere la funzione 'LRN' la riga n. 2 con la colonna 'A'.

Ci si rende subito conto delle difficoltà di manipolare i dati sotto questa forma e allo stesso tempo della macchinosità della cosa, in quanto utilizzando la tastiera a questa maniera arriveremmo al limite della praticità con un byte lungo addirittura 40 bit. Sono perciò ricorso a switches elettronici di tipo C-mos comandati da decodifiche binario-decimale attraverso opportuni adattamenti dell'informazione trattata.
Semplice eh? ma se osservate lo schema di figura 2 tutto sarà ancora più semplice e chiaro!

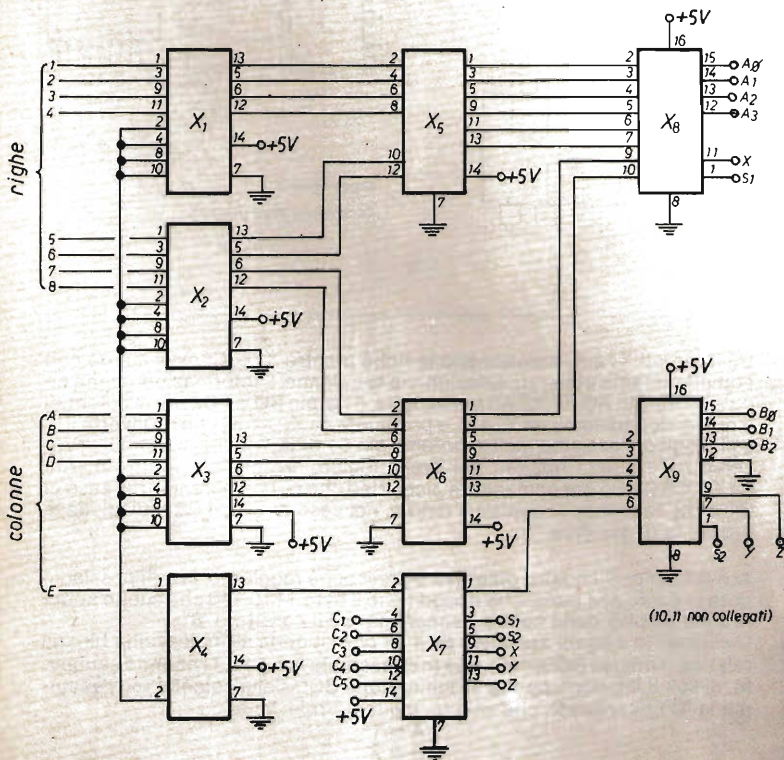
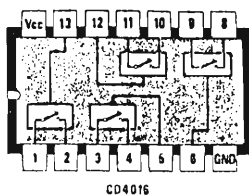
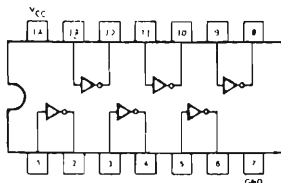
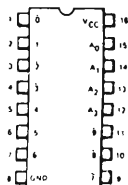
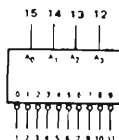



figura 2

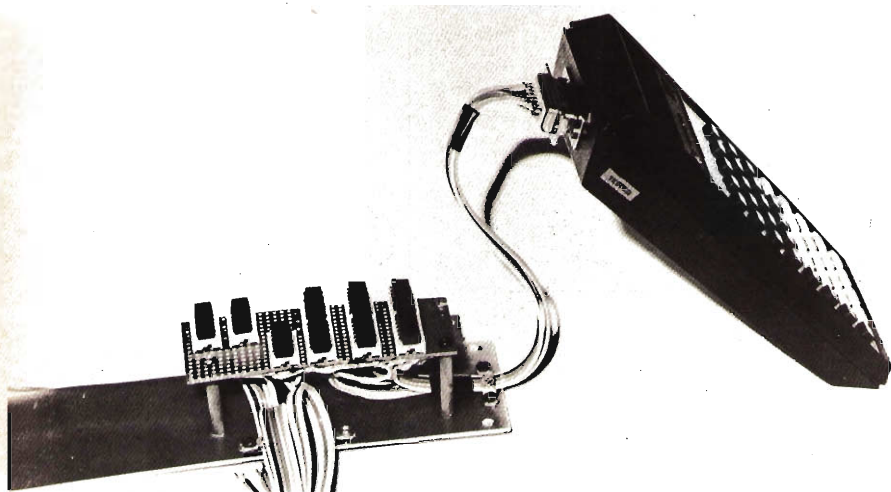
Zoccolature degli integrati a pagina seguente.

segue figura 2

Zoccolature degli integrati.

Gli integrati X_1 e X_2 selezionano le righe mentre X_3 e X_4, ovviamente non completamente utilizzati, selezionano le colonne. L'informazione viene inputtata dai pin AØ,A1,A2,A3 per le righe, e dai pin BØ,B1,B2 per le colonne; è quindi decodificata da X_8 e X_9 e preparata da X_5, X_6, X_7 che ne inverte il livello logico. Possiamo inoltre notare i pin S1,S2,X,Y,Z che, transitati attraverso X_7 e quindi ribaltati nel livello logico, diventano rispettivamente C1,C2,C3,C4,C5 per comandare piccoli led che ci indicheranno, nel caso di C1 e C2, l'assenza di qualsiasi dato e, nel caso di C3,C4,C5 attivati, della presenza di overflow.

Arrivati sin qui, il nostro obiettivo è pressoché raggiunto: infatti possiamo dare in pasto alle fameliche decodifiche il byte 1110 110 che subito apparirà nel display della nostra beneamata '57 il carattere '8'.
Ovviamente il primo semibyte da 4 bit corrisponde all'indicazione binaria relativa, o meglio BCD, della riga (nell'esempio la n. 7) e il secondo semibyte, di soli 3 bit, corrisponde all'indicazione BCD della colonna (nell'esempio la 'C') ricordando che vale la seguente relazione:

A = 1
B = 2
C = 3
D = 4
E = 5

Ora entra in gioco la vostra fantasia e tutta l'ingegnosità di cui disponete; io personalmente ho utilizzato il byte così ottenuto per registrare i programmi su banda perforata leggendo l'informazione con comuni fotoresistenze

che pilotavano i relativi transistori, ma non ci si deve fermare qui: volendo si può trasformare l'informazione da parallela a seriale, registrare quindi i dati su nastro magnetico, addirittura trasmetterli e così via sino al limite della nostra immaginazione: l'importante è avere la coscienza di fare qualcosa per restare al passo di questo sviluppo tecnologico travolgente e di essere entrati in un mondo che ancora a molti fa paura, ma che sarà il pane di molti tecnici di domani.

TRUTH TABLE (C1 → 0, C3 → 9)

A_0	A_1	A_2	A_3	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$	$\bar{4}$	$\bar{5}$	$\bar{6}$	$\bar{7}$	$\bar{8}$	$\bar{9}$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
H	L	L	L	H	L	H	H	H	H	H	H	H	H
L	H	L	L	H	H	L	H	H	H	H	H	H	H
H	H	L	L	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	H	H	H	L	H	H	H	H	H
H	L	H	L	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
H	H	H	L	H	H	H	H	H	H	H	L	H	H
L	L	L	H	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

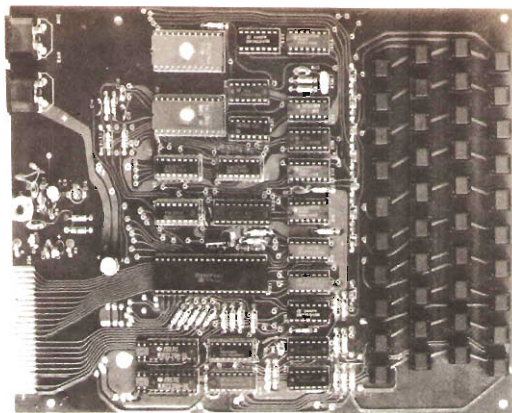
TRUTH TABLE (C2 → 0, C4 → $\bar{6}$, C5 → $\bar{7}$)

B_0	B_1	B_2	$\bar{0}$	A	B	C	D	E	$\bar{6}$	$\bar{7}$
L	L	L	L	H	H	H	H	H	H	H
H	L	L	H	L	H	H	H	H	H	H
L	H	L	H	H	L	H	H	H	H	H
H	H	L	H	H	H	L	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H
H	L	H	H	H	H	H	H	L	H	H
L	H	H	H	H	H	H	H	H	L	H
H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level

figura 3

Un ultimo sguardo ora alla figura 3 che rappresenta la tavola della verità del byte di ingresso utile per la codificazione dei caratteri usati e per l'utilizzo degli indicatori C1 (Ø nel primo semibyte), C2 (Ø nel secondo semibyte), C3 (overflow nel primo semibyte), e C4,C5 (due condizioni di overflow per il secondo semibyte).

Dalle foto, inoltre, per le quali va un particolare ringraziamento all'amico **Claudio Gatti,** notiamo i particolari del montaggio dell'interfaccia e della modifica alla TI57. ✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲

C1 DIGITAL COMPUTER

SCHEDA MICROCOMPUTER basata su microprocessore Z80/A

- Linguaggio **Basic**
- Tastiera alfanumerica - 40 tasti
- Uscita video universale
- Presentazione 32 caratteri per 24 righe

Memoria R.A.M. disponibilità 1 K
Sistema operativo su EPROM da 4 K
Entrata e uscita per registratore
Alimentazione 5 Volt stabilizzati
Connettore posteriore per future espansioni
Sono in allestimento le espansioni di memoria da 8 e 16 Kbit

CONOSCETE I NOSTRI FALCONKIT?

FK 100/C - Car stereo booster 30+30 W
FK 110/C - Antenna portabollo
FK 120/C - Led Vu-meter per auto
FK 130 - Led Vu-meter profess.
FK 140/C - Antifurto per auto
FK 150 - Sirena elettronica con altoparlante 10 W
FK 150/C - Sirena elettronica con contenitore
FK 160/C - Luci psichedeliche per auto
FK 170 - Luci psichedeliche profess.
FK 180 - Luci stroboscopiche
FK 190 - Amplificatore mono 7 W hi-fi
FK 200 - Amplificatore mono 15 W hi-fi
FK 210/C - Contagiri per auto a led
FK 220 - Orologio digitale a display giganti
FK 230 - Preamplificatore stereo hi-fi
FK 250/C - Lampeggiatore con relè
FK 260 - Metronomo elettronico
FK 270/C - Timer
FK 280/C - Alimentatore stabilizzato (utilizzabile in particolare per il C1)

N.B. - gli articoli . . . /C vengono forniti completi di contenitore

Potete trovare i nostri FALCONKIT presso tutti i migliori negozi di elettronica della Vostra citta. Saremo lieti di fornirVi i nominativi. **TELEFONATECI!**

Tutto quello che avreste voluto sapere .... sulle EPROM (e che non avete mai osato chiedere)

Paolo Sinigaglia

Non preoccupatevi, stavo scherzando; poca teoria e molti fatti (o quasi).
Tanto per non smentirmi, inizierò parlando di tutto salvo che delle EPROM.

Viene chiamato RAM (Random Access Memory cioè memoria ad accesso casuale) qualunque elemento che contenga un certo numero di circuiti bistabili in grado di contenere una cifra binaria (0 o 1) ciascuno, in cui i bistabili non siano accessibili direttamente ma solo tramite indirizzamento.
Lo so che non si capisce niente, ma non è colpa mia; mettiamola così: una RAM è un circuito a cui oltre a dire di memorizzare qualcosa, dobbiamo anche dire dove memorizzarla, cioè in quale parte della memoria.
Tutto chiaro? Bene, e le ROM? ROM non vuol dire, come qualcuno può aver pensato...Random Occess Memory, bensì Read Only Memory (memoria a sola lettura). Suona un po' reazionario una memoria che ricorda sempre il passato senza possibilità di cambiamento, ma in qualche caso (a dire il vero in molti casi) possono anche essere utili, il concetto comunque non è molto diverso da quello delle RAM, si potrebbe dire che sono delle RAM il cui contenuto viene deciso al momento della fabbricazione e non può più venir cambiato (è sconsigliabile inventare le Write Only Memory, anche se sarebbero state comode a Nixon ai tempi di Watergate...).
Dopo le ROM vengono le PROM (questo è facile e vuol dire davvero Programmable Read Only Memory) e cioè memoria a sola lettura programmabile (sarebbe come dire che si può solo leggere ma anche scrivere).
Il problema sta nel fatto che nelle PROM ogni bit, 1 o 0, è rappresentato da un fusibile all'interno dell'integrato; quando la PROM viene venduta (o comprata a seconda dei punti di vista) tutti i fusibili sono intatti; per mezzo di impulsi di tensione è possibile bruciare selettivamente alcuni dei fusibili ottenendo così la programmazione. Il vantaggio delle PROM rispetto alle ROM è che l'utente può programmarle da solo riducendo (molto) i costi per piccole quantità; rispetto alle RAM il vantaggio sta nel fatto abbastanza ovvio che, una volta programmato, il contenuto della PROM rimane inalterato anche se non è applicata l'alimentazione. Gli svantaggi sono due: la programmazione delle PROM è una faccenda lenta e, essendo chiaramente impossibile ricostruire un fusibile bruciato lungo un decimo di milllimetro, è anche impossibile correggere un errore di programmazione.

E le EPROM? direte voi

Calma, adesso ci arrivo.
EPROM sta per Erasable Programmable Read Only Memory cioè memoria a sola lettura, scrittura e cancellazione (si, sono d'accordo, è una cosa assurda ma non è colpa mia).
Nelle EPROM ogni cella di memoria è costituita da un mosfet in cui, oltre a un gate "normale" (vedi figura 1), esiste un gate "fluttuante".

STORAGE CELL CROSS SECTION

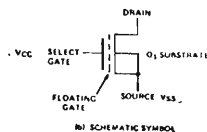
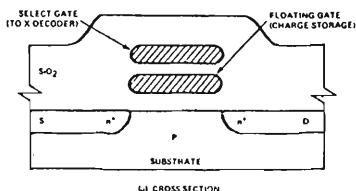



figura 1

Sezione trasversale del mosfet che costituisce la cella di memoria di una EPROM (a sinistra) e simboli della EPROM (a destra).
Al centro della sezione si vedono il gate "normale" e quello fluttuante, indicato tratteggiato nella rappresentazione schematica.

La programmazione viene ottenuta imprigionando delle cariche negative (elettroni nella fattispecie) nel gate fluttuante per mezzo di impulsi elettrici.

STORAGE CELL THRESHOLD SHIFT

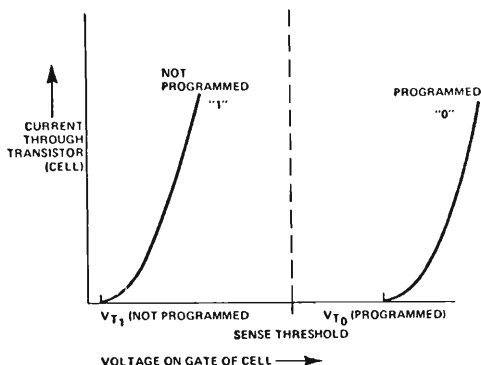



figura 2

Grafico della corrente circolante nel mosfet al variare della tensione applicata al gate "normale".
Si veda come la tensione di soglia nella cella non programmata (V_{T_1}) sia molto inferiore che non nella cella programmata (V_{T_0}).

La presenza delle cariche nel gate fluttuante modifica le caratteristiche del mosfet nel modo indicato in figura 2; la tensione di threshold, la tensione minima da applicare al gate normale per ottenere la conduzione, passa da un valore nettamente inferiore a quello indicato come sense threshold a uno superiore. Applicando al gate la tensione di sense threshold è possibile stabilire, in base alla conduzione del canale, se la cella in questione è programmata o meno. Poiché il gate fluttuante è circondato da SiO_2 che è un ottimo isolante, le cariche possono rimanere intrappolate per tempi molto lunghi. In effetti l'unico modo con cui è possibile liberare le cariche dal gate fluttuante è fornire loro un'energia sufficiente a superare la barriera costituita dall'ossido di silicio. Questa energia può venire fornita mediante l'illuminazione con luce della stessa lunghezza d'onda, di solito nella banda dell'ultravioletto. per dovere di cronaca devo dire che esistono anche dei circuiti detti EEPROM o E^2PROM che sta per Electrically Erasable eccetera, Cioè PROM concellabili elettricamente, che assomigliano molto alle EPROM ma che possono essere cancellate mediante impulsi elettrici invece che con l'ultravioletto. Si tratta di circuiti costruiti con una tecnologia relativamente nuova e non sono ancora riuscito a saperne quasi niente; pazienza, non dum matura est.
Ma allora come si usano queste EPROM? Io propongo di vedere separatamente i seguenti punti: come si fa a leggere, a cancellare, a programmare le EPROM più diffuse e infine quali circuiti usare per farlo. Sia ben chiaro che questa scaletta è solo una mia proposta e voi siete padronissimi di non essere d'accordo con questa suddivisione, non che la cosa mi interessi molto, anzi, a dirla tutta, non mi interessa affatto. Quindi, per dimostrare che non mi interessa neanche la scaletta, incomincio subito dal secondo punto e cioè dalla CANCELLAZIONE o, come dicono quelli che hanno studiato, ERASING.
Cancellare una EPROM è la cosa più facile del mondo, quello che è più difficile è impedire che si cancelli da sola...
In figura 3 avete sotto il naso un grafico del tempo di cancellazione in funzione della lunghezza d'onda a parità di illuminazione.

N-CHANNEL EPROM PHOTO-ERASE CHARACTERISTIC

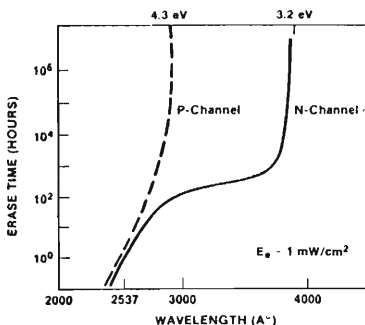



figura 3

Il tempo di esposizione necessario a cancellare una EPROM a canale N (tratto continuo) o a canale P (tratteggiato) è qui riportato in funzione della lunghezza d'onda della luce usata a parità di illuminazione. Anche se ovviamente non si può vedere dal grafico, il tempo di esposizione è inversamente proporzionale all'illuminazione.

Il grafico tratteggiato vale per le EPROM a canale P mentre l'altro vale per quelle a canale N. Prenderemo in esame solo le EPROM a canale N perché sono le più diffuse (ma esistono anche le altre). Vediamo dal grafico che il tempo di cancellazione aumenta all'aumentare della lunghezza d'onda fino a circa 3.800Å, per λ superiori il tempo necessario aumenta molto rapidamente e, per λ > 3.900Å circa non si ha cancellazione qualunque sia il tempo di esposizione e l'illuminazione (è comunque sconsigliabile cercare di verificarlo mettendo la EPROM in un forno a infrarossi); il tempo d'esposizione è invece inversamente proporzionale all'illuminazione (che a sua volta è inversamente proporzionale al quadrato della distanza dalla sorgente di luce). L'ovvia deduzione è che per cancellare è necessario usare una lampada che emetta luce con λ < 3.800Å. Il modo migliore per ottenerla è di usare una lampada a vapori di mercurio non filtrata (sono inutili a questo scopo le lampade cosiddette "solari" o quelle a "luce nera" poiché non emettono quantità sufficiente di ultravioletto lontano), vanno invece benissimo le lampada germicide usate per la sterillizzazione degli strumenti chirurgici.
ATTENZIONE: l'ultravioletto lontano è pericoloso; un'esposizione prolungata può provocare tumori alla pelle, guardare direttamente la lampada in funzione può dare danni irreparabili agli occhi. È necessario montare la lampada in un contenitore chiuso in modo che si spenga se viene aperto. Adesso che ho compiuto il dovere di mio bravo cittadino avvertendovi dei rischi, posso tirare avanti.
Il prossimo problema è accendere la lampada, non starò a dilungarmi e vi lascio alle prese con figura 4 (è un po' sibillina ma non difficile).

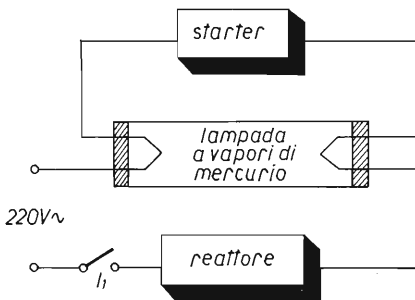



figura 4

Se la lampada, il reattore e lo starter sono interi e se c'è tensione si dovrebbe accendere. I_1 è l'interruttore di sicurezza per chi si preoccupa dei propri occhi.

Il prossimo è più complicato: per quanto tempo bisogna lasciare l'EPROM sotto la lampada? Per avere una cancellazione completa con una lampada a vapori di Hg è necessario avere una densità di energia di 15 Ws/ cm^2 il che significa che, se la lampada è cilindrica di potenza P, lunghezza L e la EPROM (o le EPROM) è a distanza D dal suo asse, il tempo T necessario alla cancellazione è dato dalla formuletta:

$$T = \frac{30 \cdot \pi \cdot L \cdot D}{P}$$

dove L e D sono espressi in centimetri, P in watt e T in secondi(ricordo che π=3,14... bisogna proprio dirvi tutto?). Prima dicevo che il problema non è stato cancellare le EPROM quanto evitare che lo facciano da sole; forse ho esagerato un po', ma il problema esiste davvero; da figura 5 si vede che, sia la luce solare, sia quella dei normali tubi fluorescenti contengono radiazioni con lunghezza d'onda inferiore alla soglia dei 3.800 Å. Questo significa che se non si copre (ad esempio con un pezzo di scotch) la finestrella delle EPROM, a lungo andare possono venir cancellate dalla luce esterna. A titolo puramente indicativo, il tempo di cancellazione con un'esposizione diretta ai raggi solari è dell'ordine delle ore, con le lampade fluorescenti è di qualche giorno. Anche se la finestrella è ben coperta, le EPROM non tengono i dati all'infinito per agitazione termica, o per azione della radioattività di fondo o dei raggi cosmici; sembra poco, e infatti è poco: secondo i dati della INTEL sono necessari mediamente dieci anni perchè una EPROM tenuta in condizioni normali perda dei dati.

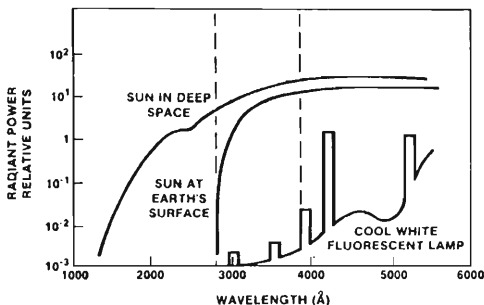



figura 5

Contenuto spettrale della luce solare e della luce delle normali lampade fluorescenti. Come si vede, una parte di ambedue ha λ inferiore ai 3800 Å e può quindi causare la cancellazione di una EPROM.

Passiamo ora al secondo punto (che era poi il primo) e cioè la lettura dei dati programmati nella EPROM; facciamo prima un discorso generale e passeremo poi a esaminare in dettaglio le due famiglie di EPROM attualmente più diffuse: la serie 2704/ 2708 e la serie 2758/ 2716/ 2732 della INTEL ed equivalenti.
Si può dire che, in generale, i terminali di una EPROM si possono dividere (salvo che durante la programmazione) in quattro gruppi: terminali di alimentazione, di abilitazione, di address (indirizzo) e di output (uscita). Per quanto riguarda l'alimentazione, in generale i terminali sono indicati con una sigla che comincia per "V", di solito V_{cc} indica l'alimentazione a 5 V positivi e V_{ss} la massa; queste sono le uniche tensioni di alimentazione necessarie per le EPROM della famiglia della 2716; per quelle della famiglia della 2708 e per la maggior parte delle altre, sono necessarie diverse altre tensioni (−5, +12, −12 V, eccetera) di alimentazione. Per alcuni di questi circuiti è NECESSARIO che le alimentazioni siano applicate e rimosse simultaneamente per evitare di fare un arrosto di eprom (di cui vanno parti-

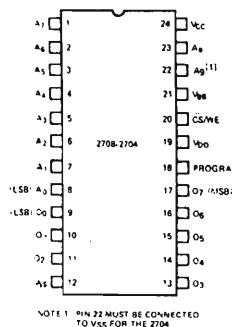
colarmente ghiotti i Rivenditori...). Saltiamo per un attimo i terminali di abilitazione e passiamo a quelli di address. Questi ultimi sono normalmente indicati come A0, A1, A2, A3 eccetera; a cosa servono? La cosa è semplice (abbastanza): servono a dire alla EPROM quale delle celle di memoria il numero binario ottenuto scrivendo uno dopo l'altro i livelli logici che si devono applicare ai terminali di address per selezionare quella data cella di memoria.
Facciamo un esempio che è meglio: consideriamo una EPROM che abbia quattro ingressi di address; la cella indirizzata dando agli address i seguenti livelli logici:

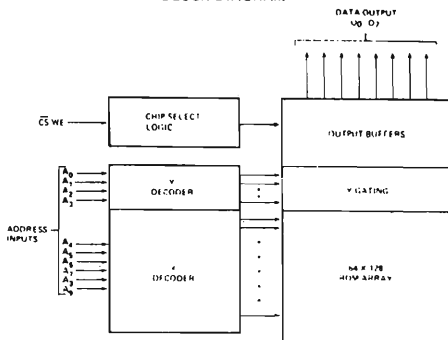
A0 = 1
A1 = 0
A2 = 1
A3 = 0

sarà indicata come cella numero 5 poiché scrivendo nell'ordine i livelli di A3, A2, A1, A0 otteniamo 0101 che in binario equivale a 5. Sono tentato di non dirvi nemmeno cosa sono i terminali di output poiché mi sembra alquanto ovvio; sono normalmente indicati come O1, O2, O3, eccetera (l'avevate gia indovinato vero?); quando la EPROM è abilitata, il livello logico degli Output ci da il contenuto della cella indirizzata.
Bene, mi sembra che fino a qui sia tutto chiaro, il prossimo punto sarà invece un po'più impegnativo; infatti è abbastanza difficile generalizzare per ciò che riguarda i terminali di abilitazione perché il loro funzionamento varia tra i vari tipi di EPROM; esamineremo quindi nei dettagli le caratteristiche di due soli circuiti: la 2716 e la 2708, per ciò che riguarda gli altri integrati di queste famiglie le cose vanno ancora bene perché le caratteristiche sono quasi identiche, per gli altri bisogna arrangiarsi per analogia. Spero che siate d'accordo sul fatto che una descrizione dettagliata di tutti i tipi di EPROM esistenti sul mercato richiederebbe un libro piuttosto che un articolo.
In figura 6 abbiamo le prime pagine dei Data Sheets della 2708 e della 2716; esaminiamo prima figura 6a: nella 2708, oltre ai piedini di address e di output, abbiamo dei piedini indicati come V_{ss}, V_{cc}, V_{bb} e V_{dd}, PROGRAM e CS/ WE. Per le alimentazioni la cosa è semplice: V_{ss} va a massa, V_{cc} a +5 V, V_{bb} a −5 V e V_{dd} a +12 V; dalla tabella riportata nella stessa figura si vede che, in condizione di lettura, anche il piedino PROGRAM va connesso a massa. È rimasto il piedino CS/ WE che permette l'abilitazione e la disabilitazione della memoria: quando CS/ WE è a livello logico basso la memoria è abilitata, quando è alto i pin di output sono in condizione di alta impendenza (Hi-Z) e cioè si comportano **quasi** come se fossero scollegati (dalla tabella delle caratteristiche in figura 7a apprendiamo che la corrente di fuga (Output Leakage Current) I_{LO} massima è di 10 μA).
D'accordo, adesso vedo di andare più svelto... Per la 2716 (figura 6b) le cose sono simili salvo che abbiamo un'unica tensione di alimentazione a 5 V che, in lettura, va applicata ai pin V_{cc} e V_{pp} e che esiste la possibilità, mettendo a livello alto il pin PD/ PGM, di entrare nello stato "power down" in cui (figura 7b) la corrente di picco assorbita scende da 105 mA a 30 mA e in cui gli output sono in Hi-Z.
Bene, ci siete ancora tutti? Speriamo...
Tutto quello che ho detto fino ad ora è vero (chi mai potrebbe dubitarne), ma si può applicare solo in condizioni statiche, cosa succeda **nel momento in cui** applichiamo gli stati logici ai piedini di address e di abilitazione non lo so e non c'è modo di saperlo; infatti, come si vede in figura 8, è necessa-

PIN CONFIGURATION

BLOCK DIAGRAM

a

PIN NAMES

A_0-A_7	ADDRESS INPUTS
O_1-O_8	DATA OUTPUTS/INPUTS
CS/WE	CHIP SELECT/WRITE ENABLE INPUT

PIN CONNECTION DURING READ OR PROGRAM

MODE	PIN NUMBER: DATA I/O 9-11, 13-17	ADDRESS INPUTS 1-8, 22, 23	V_{SS} 12	PROGRAM 18	V_{DD} 19	CS/WE 20	V_{BB} 21	V_{CC} 24
READ	D_{OUT}	A_{IN}	GND	GND	+12	V_{IL}	-5	[illegible]
DESELECT	HIGH IMPEDANCE	DON'T CARE	GND	GND	+12	V_{IH}	-5	[illegible]
PROGRAM	D_{IN}	A_{IN}	GND	PULSEC 26V	+12	V_{IHW}	-5	[illegible]

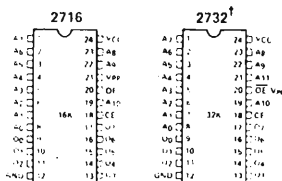
utilissimo

PIN CONFIGURATION

MODE SELECTION

PINS / MODE	$\overline{CE}$/PGM (18)	$\overline{OE}$ (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Standby	V_{IH}	Don't Care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

BLOCK DIAGRAM

b

PIN NAMES

A_0-A_{10}	ADDRESSES
$\overline{CE}$/PGM	CHIP ENABLE/PROGRAM
$\overline{OE}$	OUTPUT ENABLE
O_0-O_7	OUTPUTS

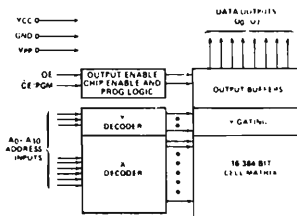



figura 6

Prima pagina dei Data Sheets delle EPROM 2708 (a) e 2716 (b) della INTEL.
Dalla figura si ricavano collegamenti, alimentazioni e schema di principio di queste EPROM che ho scelto come rappresentative delle rispettive famiglie, certamente tra le più diffuse.

Absolute Maximum Ratings*

Temperature Under Bias . . . . -25°C to +85°C
Storage Temperature . . . . -65°C to +125°C
V_{DD} With Respect to V_{BB} . . . . +20V to -0.3V
V_{CC} and V_{SS} With Respect to V_{BB} . . . . +15V to -0.3V
All Input or Output Voltages With Respect
to V_{BB} During Read . . . . +15V to -0.3V
$\overline{CS}$/WE Input With Respect to V_{BB}
During Programming . . . . +20V to -0.3V
Program Input With Respect to V_{BB} . . . . +35V to -0.3V
Power Dissipation 1.5W

NOTE:
1) Vbb deve essere applicata per prima e rimossa per ultima.
2) Valori tipici per T_A=25°C
3) La dissipazione totale non può essere calcolata moltiplicando le tensioni di alimentazione per le correnti massime

DC and AC Operating Conditions During Read

	2708	2708-1	2708L
Temperature Range	0°C – 70°C	0°C – 70°C	0°C – 70°C
V_{CC} Power Supply	5V ± 5%	5V ± 5%	5V ± 10%
V_{DD} Power Supply	12V ± 5%	12V ± 5%	12V ± 10%
V_{BB} Power Supply	-5V ± 5%	-5V ± 5%	-5V ± 10%

a

READ OPERATION
D.C. and Operating Characteristics

Symbol	Parameter	2708, 2708-1 Limits			2708L Limits			Units	Test Conditions
		Min.	Typ.[2]	Max.	Min.	Typ.[2]	Max.		
I_{LI}	Address and Chip Select Input Sink Current		1	10		1	10	µA	V_{IN} = 5.25V [illegible]
I_{LO}	Output Leakage Current		1	10		1	10	µA	V_{OUT} = 5.5V, CS/WE = 5V
I_{DD}[3]	V_{DD} Supply Current		50	65		21	28	mA	Worst Case Supply Currents[4]
I_{CC}[3]	V_{CC} Supply Current		6	10		2	4	mA	All Inputs High
I_{BB}[3]	V_{BB} Supply Current		30	45		10	14	mA	$\overline{CS}$/WE = 5V; T_A = 0°C
V_{IL}	Input Low Voltage	V_{SS}		0.65	V_{SS}		0.65	V	
V_{IH}	Input High Voltage	3.0		V_{CC}+1	2.7		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45			0.4	V	[illegible] 2708, 2708-1 / [illegible] 2708L
V_{OH1}	Output High Voltage	3.7			3.7			V	I_{OH} = [illegible]
V_{OH2}	Output High Voltage	2.4			2.4			V	[illegible]
PD	Power Dissipation			800			325	mW	T_A = [illegible]
							425	mW	T_A = 0°C

Absolute Maximum Ratings*

Temperature Under Bias . . . . -10°C to +80°C
Storage Temperature . . . . -65°C to +125°C
All Input or Output Voltages with
Respect to Ground . . . . +6V to -0.3V
V_{PP} Supply Voltage with Respect
to Ground During Program . . . . +26.5V to -0.3V

NOTE:
1) Vcc deve essere applicato prima di Vpp e rimosso dopo (o contemporaneamente)
2) La corrente totale è Vcc+Vpp_1
3) Valori tipici per T_A=25°C

DC and AC Operating Conditions During Read

	2716	2716-1	2716-2
Temperature Range	0°C – 70°C	0°C – 70°C	0°C – 70°C
V_{CC} Power Supply[1,2]	5V ± 5%	5V ± 10%	5V ± 5%
V_{PP} Power Supply[2]	V_{CC}	V_{CC}	V_{CC}

b

READ OPERATION

D.C. and Operating Characteristics

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ.[3]	Max.		
I_{LI}	Input Load Current			10	µA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	µA	V_{OUT} = 5.25V
I_{PP1}[2]	V_{PP} Current			5	mA	V_{PP} = 5.25V
I_{CC1}[2]	V_{CC} Current (Standby)		10	25	mA	$\overline{CE}$ = V_{IH}, $\overline{OE}$ = V_{IL}
I_{CC2}[2]	V_{CC} Current (Active)		57	100	mA	$\overline{OE}$ = $\overline{CE}$ = V_{IL}
V_{IL}	Input Low Voltage	-0.1		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -400 µA

figura 7

Seconda pagina dei Data Sheets; e vi ho anche tradotto le note!

rio un certo tempo perché dopo l'applicazione degli indirizzi i dati in uscita siano validi (T_{ACC} = 450 ns max) o perché dopo che CS/ WE sia diventato basso gli output escano dallo stato Hi-Z (T_{CO} = 120 ns max) e, ancora, per la 2716 è necessario un certo tempo (T_{ACC_2} = 450 ns max) per uscire dallo stato di power down(o Standby che è lo stesso).

A. C. Characteristics

Symbol	Parameter	2708-1 Limits Min.	Typ.	Max.	2708, 2708L, Limits Min.	Typ.	Max.	Units
t_{ACC}	Address to Output Delay		280	350		280	450	ns
t_{CO}	Chip Select to Output Delay		60	120		60	120	ns
t_{DF}	Chip Deselect to Output Float	0		120	0		120	ns
t_{OH}	Address to Output Hold	0			0			ns

CAPACITANCE[1] T_A = 25°C, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Unit.	Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

Note: 1. This parameter is periodically sampled and is not 100% tested.

A.C. TEST CONDITIONS:

Output Load: 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times: ≤20 ns
Timing Measurement Reference Levels: 0.8V and 2.8V for inputs; 0.8V and 2.4V for outputs.
Input Pulse Levels: 0.65V to 3.0V

Waveforms

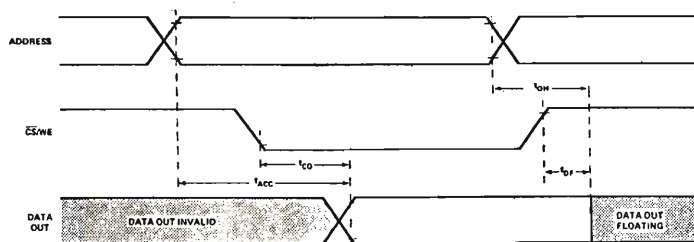



figura 8

Caratteristiche della 2708 in corrente alternata (in lettura); per la 2716 le caratteristiche sono uguali alla 2708L, per la 2716-1 sono uguali alla 2708.

Abbiamo così finito di parlare di come si legge una EPROM, a questo punto si impone un doppio salto condizionato: se avete capito tutto fino a qui, prendetevi un attimo di respiro dalla mattonata e leggete il seguito su **cq elettronica** *di* **aprile;** *in caso contrario, se pensate che la colpa è mia, piantate tutto lì e passate al prossimo articolo così risparmiate tempo (il tempo è denaro e il denaro sono soldi), se invece pensate che la colpa sia vostra allora dovete leggere da capo (se pensate che ne valga la pena).* ✲✲✲✲✲

Bozza di progetto
per un
VFO computerizzato

ing. Gianni Becattini

Sul numero 11/81 di **cq elettronica** *è comparso un interessante articolo sui generatori di frequenza PLL, encomiabile per la sua chiarezza.*
La lettura di questo scritto mi ha fornito lo spunto per un progetto di massima di un VFO controllato a microprocessore. Anche se si tratta di uno studio essenzialmente teorico, l'analisi preventiva delle varie sezioni è stata ugualmente molto curata e sono certo che chiunque lo desideri possa facilmente tradurre in pratica questo interessante circuito.

Descrizione funzionale del prodotto finito

L'ipotetico VFO è racchiuso in un contenitore di piccole dimensioni.
Il frontale potrebbe essere quello di figura 1.

per OM

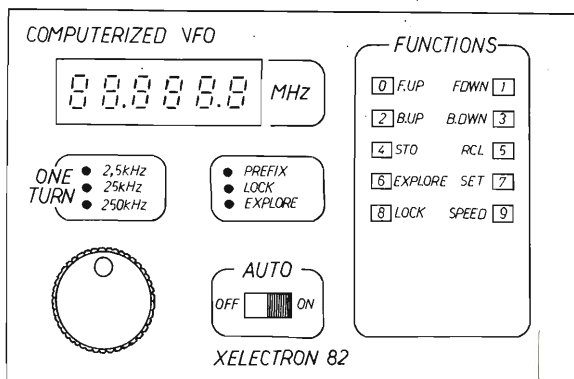



figura 1

Sul pannello abbiamo alcuni led, un display, alcuni pulsanti, un deviatore, oltre l'interruttore di accensione, e una manopola per la sintonia.
I tasti, oltre che per le funzioni, servono anche per la impostazione di quantità numeriche.
Esaminiamoli nel dettaglio:
F UP (Frequency up)-Incrementa le cifre sul display e la frequenza di uscita del VFO. La velocità con cui avviene la variazione è dapprima lenta e poi via via crescente. Col deviatore AUTO in posizione ON la ricerca continua automaticamente anche quando si abbandona il tasto e si arresta quando perviene un segnale dal ricevitore.
F DWN (Frequency down)-Come il precedente ma a decrementare.
B UP, B DWN (Band up, Band down) come i precedenti ma con azione sulle sole cifre più alte del display, per effettuare la commutazione di gamma.
STO (Store)-Permette di memorizzare una di 99 frequenze a piacere. Si preme STO seguito dal numero della memoria corrispondente. Le memorie da 0 a 9 sono indicate come 00, 01,....09. L'impostazione dei numeri avviene con gli stessi tasti funzione.
RCL (Recall)-Si usa come il precedente e serve per richiamare una qualsiasi delle frequenze precedentemente memorizzate. Con alcuni accorgimenti circuitali, che vedremo dopo, è possibile memorizzare oltre alla frequenza anche il modo di ricezione (AM/ FM/ LSB ecc.).
EXPLORE-Esplora la banda, in su e in giù, compresa tra i valori di frequenza registrati nelle memorie 01 e 02. Se è stato inserito l'AUTO si ha l'arresto per circa due secondi su ogni stazione incontrata nella esplorazione. L'EXPLORE viene disattivato premendo un tasto qualunque.
SPEED-Viene usato per variare il rapporto di demoltiplica della manopola rotante e deve essere seguito da 1, 2 o 3 che corrispondono a:

1 - un giro = 2.500 Hz (step 100 Hz)
2 - un giro = 25 kHz (step 1.000 Hz)
3 - un giro = 250 kHz (step 10 kHz)

Se invece di battere 1, 2 o 3 si batte SET, una serie di cifre e SET di nuovo è possibile prefissare uno step diverso (entro limiti preassegnati) e tutti i led dell'indicatore di rapporto (i tre led segnati ONE TURN=) si spengono.
LOCK-Disabilita la tastiera e la manopola rotante finché non si preme ancora LOCK.
SET-Consente lo spostamento su una frequenza assegnata da tastiera. Si batte SET, la frequenza desiderata, che comparirà sul display, e SET di nuovo.

I led indicano:
LOCK - che la tastiera e la manopola sono disabilitate;
EXPLORE - che il VFO sta compiendo esplorazioni;
PREFIX - che è stato premuto un tasto dietro al quale si attendono altri tasti (STO - RCL - SET - SPEED);
ON TURN= - indica il rapporto di demoltiplica scelto per la manopola.
Tutti i led spenti indicano un rapporto non standard introdotto manualmente da tastiera.

Il circuito

Il circuito è molto tradizionale: uno Z-80 provvede comodamente a tutte le funzioni, come si può osservare dallo scherma a blocchi (figura 2).
La parte PLL e divisore programmabile non è stata studiata nel dettaglio, esulando dall'oggetto proposto: non mancano tuttavia ottimi esempi sulle pagine di **cq elettronica.**

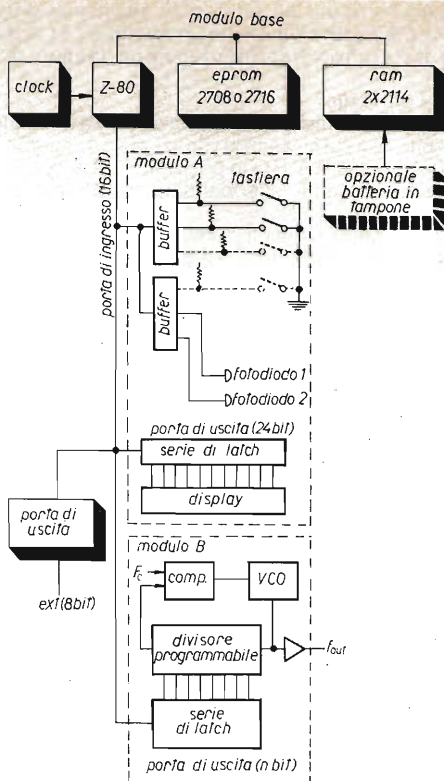



figura 2

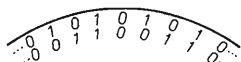
I riquadri tratteggiati indicano i moduli fisici di circuito stampato. Tutto il programma dovrebbe stare largamente in un solo k e si può usare quindi una 2708 come eprom (meglio se del tipo a una sola alimentazione). Due 2114 compongono invece la memoria RAM che viene quindi anche essa ad essere di 1 k, più che sufficiente anche per espansioni future eventuali. La batteria in tampone non è necessaria, potendosi ricorrere alla soluzione più semplice di prelevare una parte della tensione a monte dell'interruttore generale; in tal modo ovviamente la ritenzione dei dati è limitata al fatto che la spina sia inserita nella presa rete e che la tensione di questa ultima non venga a mancare. La soluzione più elegante, anche se più costosa, è quella di usare una memoria a basso consumo e una batteria in tampone.
Le porte di uscita possono essere ottenute semplicemente con degli economici latch e quelle d'ingresso con dei tri-state buffer. Ne esistono tipi a basso consumo che possono essere montati direttamente sul bus senza bufferizzazioni.

Dato che tanto la memoria quanto l'I/ O non sono molto estesi, non è necessaria una completa decodifica dell'address bus per le selezioni.

Il comando rotativo

Il comando rotativo della sintonia è ispirato a quello comparso su **cq elettronica** (9/ 80).
La manopola fa girare un disco che reca dei fori alla periferia per intercettare un raggio luminoso diretto a un fotodiodo. Nell'articolo citato si ricorreva a un motorino in cc montato sullo stesso albero per identificare il senso del moto. Il nostro progetto di VFO tenta invece una strada diversa. Le coppie led/ fotodiodo sono due invece di una e due sono le file di fori disposte come figura 3.

figura 3

La foratura del disco ottico.
0 = non foro
1 = foro

Si puo constatare che facendo ruotare il disco in una certa direzione i led si accendono secondo la sequenza:

.../ .
00
01
10
11
.../ .

e viceversa per la direzione opposta.
È quindi come se avessimo un contatore binario a due bit che conta da 0 a 3 rispettivamente per rotazioni in un certo senso (che fisseremo che sia quello orario) e da 3 a 0 per il verso opposto. Ecco quindi che diventa possibile, a seconda dell'ordine crescente o decrescente delle combinazioni, risalire alla direzione del moto. Il problema è rappresentato dai passaggi da 11 a 00 e viceversa: il contatore ha ad esempio uscite 0, 1, 2, 3, 0, 1, 2.. e la sequenza crescente si arresta per ricominciare da zero ogni volta. Questi "ritorni indietro" o "salti" del conteggio possono tuttavia essere facilmente interpretati dal programma.

Facciamo alcune considerazioni.
Chiamiamo:

L_t il valore della lettura attuale dei led;
L_{t-1} il valore della lettura precedente;

Paragonando L_t e L_{t-1} posso determinare:
a) se il disco è fermo;
b) se il disco si muove;
c) se il disco si muove in avanti;
d) se il disco si muove indietro;
e) se si è verificato un errore: ad esempio se si è passati dalla posizione 1 alla 3 senza passare dalla 2. Ciò può avvenire per vari motivi come ad esempio una rotazione troppo veloce della manopola; come vedremo nel seguito è comunque una situazione che non si dovrebbe mai verificare.

Il caso a) è il più banale; il disco è fermo se $L_t = L_{t-1}$.
I casi b), c), d) possono essere risolti con l'algoritmo riportato in figura 4.

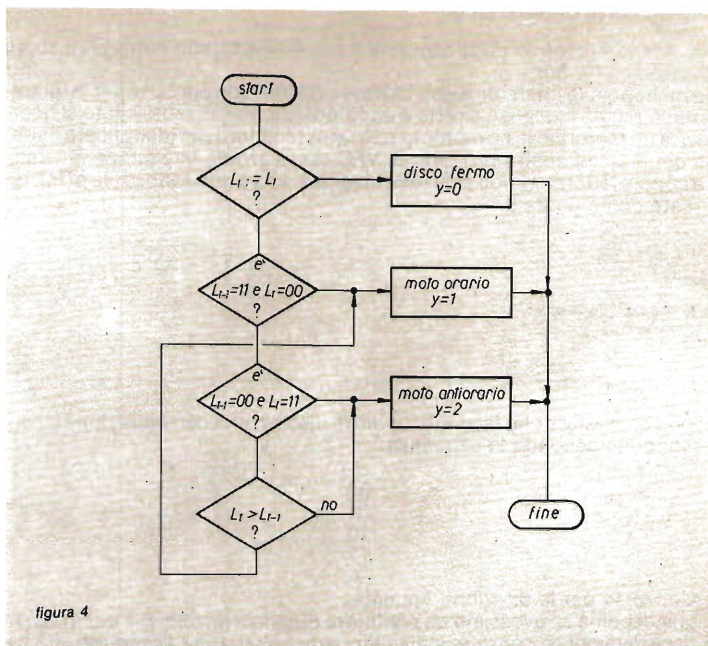



figura 4

Sul caso e) non ci soffermiamo per ora.

Ho provato a scrivere diversi programmi in Assembler Z-80 prima di arrivare a quello che sto per illustrare che è certo il più veloce e il più interessante ai fini didattici e che vale quindi la pena di studiare nel dettaglio.

Il programma MOTO

Il programma non deve fare altro che risolvere la semplice funzione

$$y = f(a,b,c,d)$$

dove a,b,c,d sono le due coppie di valori L_t L_{t-1} e y può assumere i valori:

y = 0 = disco fermo;
y = 1 = disco avanti;
y = 2 = disco indietro;
y = 3 = errore.

Invece di fare un programma che analizzi tutte le possibilità, si preferisce invece creare una tabella a 16 posti (i quattro ingressi a,b,c,d danno appunto 16 combinazioni possibili) dove per ogni combinazione possibile è registrata a lato la relativa y.
La tabella è riportata in figura 5.

lettura precedente bit		*lettura attuale bit*		*y*	*stato*
1	*0*	*1*	*0*		
0	*0*	*0*	*0*	*0*	*FERMO*
0	*0*	*0*	*1*	*1*	*AVANTI*
0	*0*	*1*	*0*	*3*	*ERRORE*
0	*0*	*1*	*1*	*2*	*INDIETRO*
0	*1*	*0*	*0*	*2*	*INDIETRO*
0	*1*	*0*	*1*	*0*	*FERMO*
0	*1*	*1*	*0*	*1*	*AVANTI*
0	*1*	*1*	*1*	*3*	*ERRORE*
1	*0*	*0*	*0*	*3*	*ERRORE*
1	*0*	*0*	*1*	*2*	*INDIETRO*
1	*0*	*1*	*0*	*0*	*FERMO*
1	*0*	*1*	*1*	*1*	*AVANTI*
1	*1*	*0*	*0*	*1*	*AVANTI*
1	*1*	*0*	*1*	*3*	*ERRORE*
1	*1*	*1*	*0*	*2*	*INDIETRO*
1	*1*	*1*	*1*	*0*	*FERMO*

figura 5

La tabella di decodifica; è possibile processare qualsiasi situazione in tempi molto brevi e con un programma molto semplice.

Il programma MOTO (figura 6) combina assieme i 4 bit di L_t e L_{t-1} e li usa come indirizzi per ricercare la relativa y in una tabella di 16 elementi.

figura 6

LOC	OBJECT CODE	#	SOURCE STATEMENT				T STATES
		1	; IPOTETICA ROUTINE PER LA DETERMINAZIONE				
		2	; DEL SENSO DI MOTO E DEL MOTO DEL DISCO				
		3	; H - DEVE CONTENERE L'OFFSET DELLA TABELLA				
		4	; C - E' USATO PER CONSERVARE IL VALORE DELLA LETT.				
		5	; PRECEDENTE - I LED SONO SUI BIT Ø E D 1 DI PORTLED				
		6	; NON PROVATA IN PRATICA				
		7	;				
		8	MOTO:	IN	PORTLED	; LEGGI IL PORT DEI LED	11
		9		AND	A,ØFCH	; TOGLI I 6 BIT PIU' SIGNIFICATIVI	4
		0				; PERCHE' ORA NON CI SERVONO	
		1		LD	D,A	; SALVANE IL CONTENUTO	4
		2		SLA	A	; SCORRI IL CONTENUTO PER	8
		3		SLA	A	; FAR POSTO AI BIT DELLA LETT. PRIMA	8
		4		OR	A,C	; COMBINA ASSIEME LET. VECCHIA E NUOVA	4
		5		LD	L,A	; CARICA L	4
		6		LD	C,D	; PONI LA NUOVA LETTURA IN C PER I CICLI SUCC.	4
		7		LD	A,(HL)	; LEGGI DALLA TABELLA LA RISPOSTA	7
		8		RET		; CHE E' Ø = DISCO FERMO	
		9				; 1 = AVANTI 2 = INDIETRO	
		0				; 3 = ERRORE	

Progetto VFO Programma MOTO Vers. PREL. T= 400 µs per clock a 2,5 MHz

Programmatore BECATTINI Foglio 1 di 1 Data 6/11/81 Rev. A

Limiti

Per curiosità ho voluto verificare i limiti di un complesso come quello ora descritto, almeno sulla carta, e mi sono calcolato approssimativamente la massima velocità che può essere impressa alla manopola perché non si abbiano "salti", cioè condizioni di errore. Se:
N = numero di divisioni del disco;
T_g = durata di un giro;
T = intervallo di tempo tra due divisioni;
g/m = velocità massima di rotazione espressa in giri al minuto si ha che:

$$T_g = \frac{1}{\frac{g/m}{60}} = \frac{60}{g/m} \quad e \quad T = \frac{60}{g/m \cdot N} \quad \text{da cui} \quad g/m = \frac{60}{N \cdot T}$$

Con T=50 µs (la routine riportata MOTO dura circa una ottantina di colpi di clock, che per $T_{clok} = 400$ ns ($f_{clock} = 2{,}5$ MHz) equivalgono a circa 30 µs) e N=50 come suggerito nell'articolo già citato, si ottiene quindi un massimo di circa 24.000 giri al minuto*. Molti si chiederanno perché ho espresso la velocità della manopola in giri al minuto: solo per rendere più intuitiva la portata del risultato; tutti abbiamo abbastanza presente questa unità di misura perché comunemente impiegata per i motori a scoppio!

(*) Valore più che buono anche considerando, tra tolleranze e approssimazioni, di ridurlo al 25%.

Il programma

Non corre più il "lontano" 1975 e sono certo che oggi la maggior parte dei Lettori interessati può cavarsela senza aiuti per stendere il programma completo sulla base di quanto ho espresso fin qui. Anzi, potrebbe questa essere la buona occasione per iniziare a prendere pratica con i microprocessori.
Io mi limito a presentare l'algoritmo base (figura 7).

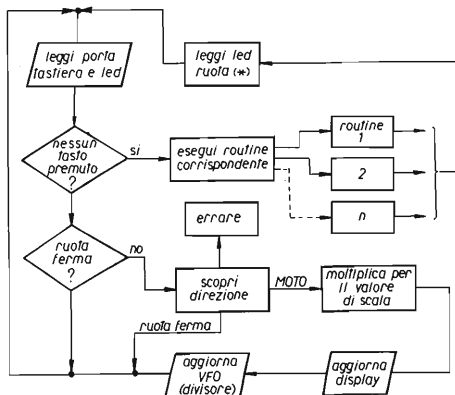



figura 7
(*) La lettura dei led è fatta per evitare segnalazioni di errore quando la manopola viene ruotata durante l'esecuzione di una funzione.

Uno dei grossi vantaggi del µp è quello di poter usare delle tabelle anche per convertire i dati del display o quelli del divisore programmabile (che di solito non usano lo stesso rango) o per effettuare qualsiasi variazione di scala.
Sullo schema elettrico è stata inserita una porta ausiliaria EXT. Il suo scopo è quello di consentire al VFO di impartire comandi alle circuiterie esterne, come a quelle del ricevitore, in funzione ad esempio della banda selezionata e del modo di ricezione quando si desideri memorizzare anche esso.

La meccanica

La meccanica risulta essere oltremodo semplice. Riporto uno schizzo di come suggerisco sia realizzata (figura 8).

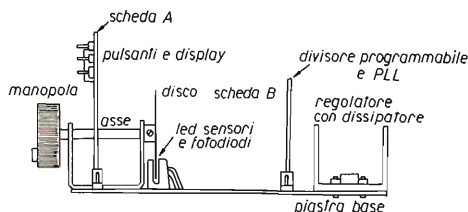
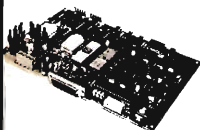



figura 8

Il disco potrebbe essere ottenuto fotograficamente: se ne guadagnerebbe in precisione. Un modo che potrei suggerire è quello di impiegare del Kodalith con la periferia a sporgere pressato tra due dischi in fibra.

Conclusione

Ritengo questa realizzazione molto interessante e mi dolgo che gli impegni quotidiani mi rendano impossibile di tradurla in pratica. Se ci fosse in zona qualche volenteroso con un po' di tempo libero per eseguire i disegni dei circuiti stampati, eventualmente con il mio aiuto, sarei ben lieto di compensarlo con tutto il materiale occorrente per costruirsi un esemplare proprio. ✻✻

GP User's Group

È stato formato un **gruppo di utilizzatori di computers General Processor mod. T,** *con lo scopo di permettere scambi di idee, consigli, applicazioni, nonché di software.*

Gli interessati possono aderire all'iniziativa compilando la scheda riprodotta qui a lato con le informazioni sul tipo di sistema posseduto e interessi circa il software, da inviarsi unitamente a una quota di lire 10.000, che servirà a coprire le spese postali e di stampa delle circolari e delle fotocopie dei manuali e istruzioni dei programmi inviati ai soci.

Per usufruire della banca software del gruppo è sufficiente spedire un disco (8") contenente i propri programmi, che andranno ad ampliare la banca. Questo verrà poi ritornato con i programmi richiesti.
È prevista la pubblicazione di un elenco dei programmi disponibili, che verrà periodicamente aggiornato.
La banca software, data l'originale impostazione, si amplierà continuamente, consentendo ai soci di usufruire di grandi quantità di programmi di ogni genere: software di base, utilità, archiviazione, statistica, matematica, varie, e anche giochi (non mancano gli scacchi).

Ricordiamo ai soci come fare per accedere alla banca programmi:

1 - prepara un disco contenente tutti i programmi che hai sviluppato o che possiedi (anche routines, utilità, ecc.); correda tutti i programmi per i quali lo ritieni opportuno, di un foglio con note, istruzioni (possibilmente fotocopiabile e scritto a macchina);

2 - spedisci il tutto al Gruppo allegando le spese postali per il ritorno e l'elenco dei programmi che vuoi ricevere (per ora, fino alla pubblicazione dell'elenco, ti verranno inviati tutti i programmi riguardanti i tuoi interessi, nei limiti di spazio del disco).

IMPORTANTE: IL SISTEMA CHE GESTISCE LA BANCA PROGRAMMI LAVORA CON DISCHI DA 8", PERTANTO I PROGRAMMI DEVONO ESSERE SPEDITI SU FLOPPY DI QUESTO FORMATO.

Stiamo cercando anche qualcuno che volesse tenere la banca su dischetti da 5", gli eventuali interessati ci scrivano al più presto.

G.P. USER'S GROUP
via CASE SPARSE, 68
34070 SAVOGNA D'ISONZO (GO)

Saremo grati a chi ci fornirà nominativi di possessori di computers GP interessati all'iniziativa.

G.P. USER'S GROUP

VIA CASE SPARSE N.68 - 34070 SAVOGNA D'ISONZO (GO)

ELABORATORE SINGOLO O
SISTEMA T-STAR O con ___ terminali

VIDEO
16 x 64 caratteri O
24 x 80 caratteri O

MEMORIA
32 K RAM O
48 K RAM O

CASSETTA O
FLOPPY 5" O n.___da___K l'uno TOT____Kbytes
FLOPPY 8" O n.___da___K l'uno TOT____Kbytes
DISCHI RIGIDI O n.___da___M l'uno TOT____Mbytes

STAMPANTE MOD.________________________________da_____car/riga
STAMPANTE MOD.________________________________da_____car/riga
PLOTTER MOD.________________________________

INTERFACCIA SIPRINT O
MODEM O tipo________________________

HARDWARE NON STANDARD____
NOTE:

SOFTWARE DI MAGGIORE INTERESSE:

	poco	abbastanza	molto
GESTIONALE	O	O	O
SCIENTIFICO	O	O	O
SOFT. DI BASE	O	O	O
UTILITA'	O	O	O
EDUCATIVO E GIOCHI	O	O	O
ALTRO (specificare): ______________________	O	O	O

NOME E COGNOME______________________________________

INDIRIZZO ______________________________________

CAP E CITTA' ______________________________________

TELEFONO ______/__________

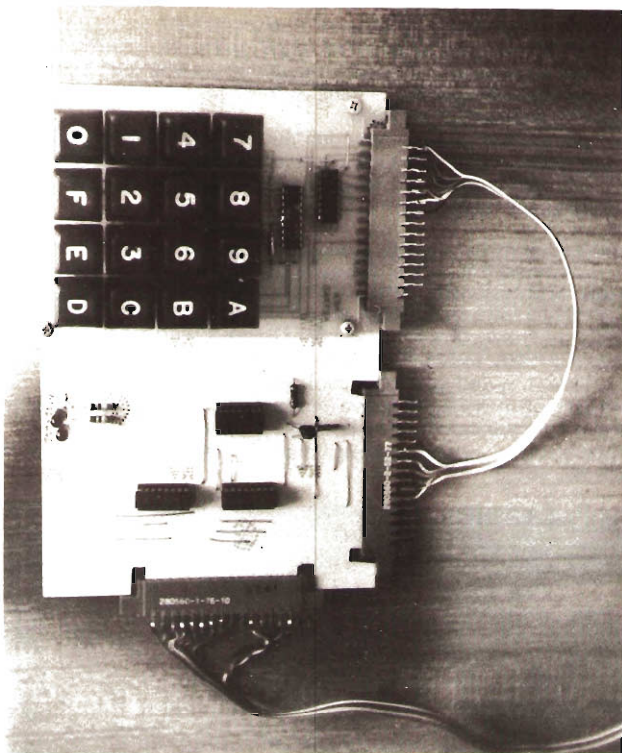
scrivere a macchina o in stampatello

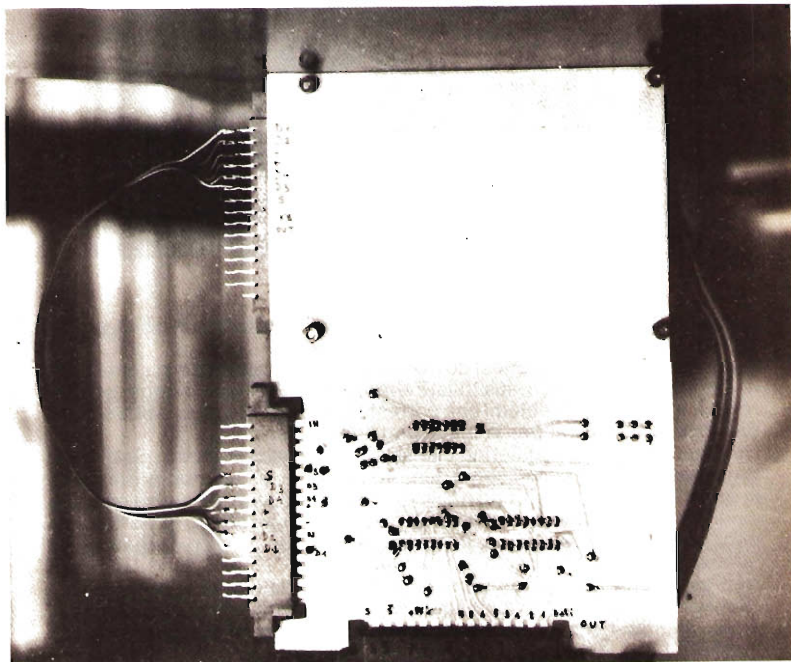
un byte
da una tastiera esadecimale

Ovvero, come usare una tastiera da un nibble per codificare un byte, correggerne a piacere i digit.
Usi: quelli che volete. Io l'ho usata con un EPROM programmer manuale.

ing. Giuseppe Aldo Prizzi

Diversi "hobbisti del personal" scrivono dei programmi di utilità per uso proprio, che poi conservano caricati in cassetta di nastro magnetico.
Diversi tra essi hanno fatto, per dirla con Marincola, il"grande passo": hanno acquistato o costruito un EPROM programmer.
Io sono tra loro.

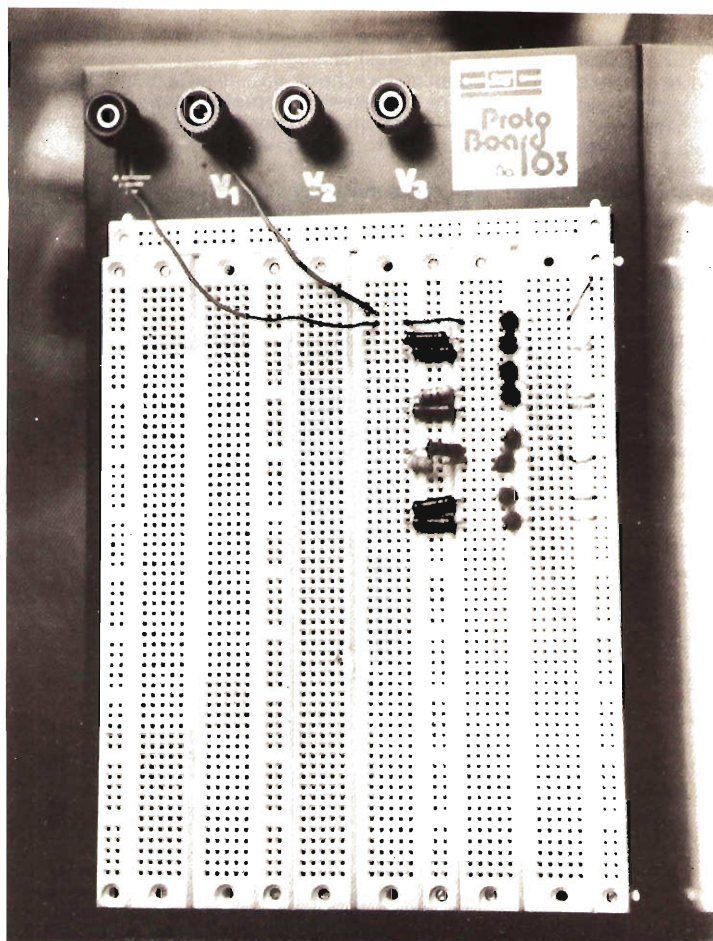
Dispongo in questo momento di due programmatori per EPROM, uno per 2516 (2 k, monoalimentazione, con impulsi di scrittura di 50 msec, positivi) asservito al micro; l'altro a programmazione manuale, costruito -come spesso succede- per avere qualcosa di provvisorio per programmare 2758 e 2532 oltre che le 2516 (2716 Texas).
Come altrettanto spesso succede, il provvisorio è diventato definitivo.
Io - in omaggio a una certa velocità di realizzazione- avevo predisposto per impostare i dati un bus di otto linee terminali ognuna su un deviatore.
In questo modo, leggendo i dati dei due nibble (1/ 2 byte) ottenuti suddividendo le otto linee in due gruppi, e trasformandoli mentalmente da binario a esadecimale, mi trovavo nella possibilità di programmare le mie EPROM come desiderato, errori di traduzione a parte.
E basta un errore per dover cancellare e riprogrammare tutta la EPROM. Credetemi, il senso di frustrazione che si prova quando alla 2046^0 cella esatta succede la 2047^0... sbagliata, è biblico. Uno pensa alle altre sedicimila manovre dei deviatori rese inutili, e cerca lo spigolo più robusto della linea Maginot, per sbattervi contro la capoccia.
Un giorno, non molto tempo fa, ero in quello stato d'animo, e avevo deciso, fermamente, di acquistare un EPROM programmer come si deve, costasse pure cifre esorbitanti. Il guaio era che cifre esorbitanti venivano chieste anche per dei programmatori non più buoni, anzi decisamente meno buoni del mio, anche nella sezione che più d'ogni altra induceva le mie crisi.
Eccomi allora a lavorare di...."testa", e il risultato lo vedete.
Ho risolto i miei problemi.
Se avete EPROM programmer di qualsiasi provenienza, purché funzionanti, e dei quali tutto sommato siete soddisfatti, fuorché, appunto, per il modo di impostare i dati, eccovi un modo di superare le vostre difficoltà: costruitevi questo semplice ed economico aggeggio!

Certamente la soluzione che vi propongo potrà sembrare un po' strana. In effetti non è la prima alla quale io sia pervenuto.
Ma per spiegarvene la genesi dovrò raccontarvi un'altra storia: come tutte le storie che racconto, è istruttiva! Permette infatti di vedere come un altro reagisce agli stessi problemi che potreste avere voi, e -magari- può suggerire altre applicazioni per la medesima idea, chissà.
Dunque. Costruita la tastierina che mi serviva -utilizzando un LSI all'uopo predisposto, e che quindi non è nemmeno il caso di commentare, almeno in questa sede (se ci saranno richieste, lo schema aspetta che valga la pena di essere descritto per dedicargli più spazio sulla rivista - e ne "varrà la pena" se le richieste saranno in numero adeguate.... ma, credetemi, non è affatto originale), mi trovo il solito amico.
Non capisco bene: io sono uno degli esseri più solitari di questo mondo. Non ho -o ne ho molto pochi- amici. È incredibile quanti si scoprano miei amici, quando hanno bisogno di qualcosa! Ma così va la vita!
Il solito amico, dicevo.

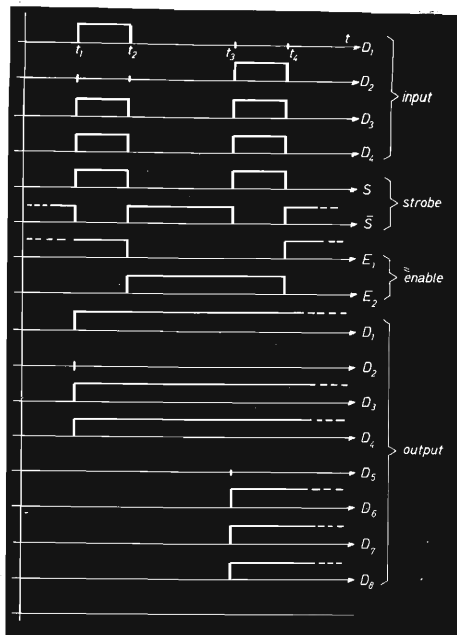
Aveva comprato una tastierina esadecimale. Anche quella per un prezzo spropositato. E voleva comprarne un'altra, per potere usare una per il nibble (in questo caso, potremmo dire il "digit") più a sinistra nella rappresentazione esadecimale del byte, l'altra per quello più a destra (come dire il digit più significativo -Most Significant Digit, MSD; e quello meno significativo -Lower Significant Digit, LSD).
Così voleva risolvere il problema del "suo" programmatore di EPROM.
Un sussulto di buon senso lo aveva portato da me, per chiedermi se potevo aiutarlo a spendere meno, e a trovare un'altra soluzione meno "stupida" di quella che aveva escogitato.
Eccola.
Una sola tastierina esadecimale, due quadrupli D-Latches, un mezzo Dual J-K FF, un transistorino, due led, qualche immondizia elettronica.
Il risultato: la pressione del primo tasto carica il MSD del byte, la successiva il LSD. Se ho fatto qualche errore, un led verde mi dice che sto caricando il MSD, uno rosso che sono sul LSD. Quindi se l'errore è nel digit più significativo, ho due possibilità.
a) mi ci trovo ora (led verde acceso): ribatto il nuovo digit;
b) sono sul digit meno significativo (led rosso acceso): ribatto lo LSD come si trova, e automaticamente mi sposto sul caso a).
Lo stesso se devo correggere il digit meno significativo (mutatis led): facilino, l'"edit", vero?

Voi capirete che l'impostazione che io ho dato, su due schede, di cui una con uno spazio vuoto per accogliere la tastiera, è "su misura" per il caso che vi ho descritto, e per tutti quelli analoghi.
Se invece volete realizzare un'unica scheda, datemi retta mantenete la disposizione separata -sia pure sullo stesso supporto- per sezione tastiera (diciamo keyboard? diciamolo pure, ma solo per non perdere l'abitudine a usare dei termini che non sono nostri, purtuttavia di uso comune), e per sezione "distribuzione dei nibble".
Io vi do' degli stampati -appunto- separati, collegati tra loro da due connettori a 15 terminali e da un "pettine" di 7 conduttori (i 4 dei dati, i 2 dell'alimentazione, quello dello "strobe").

Come funziona

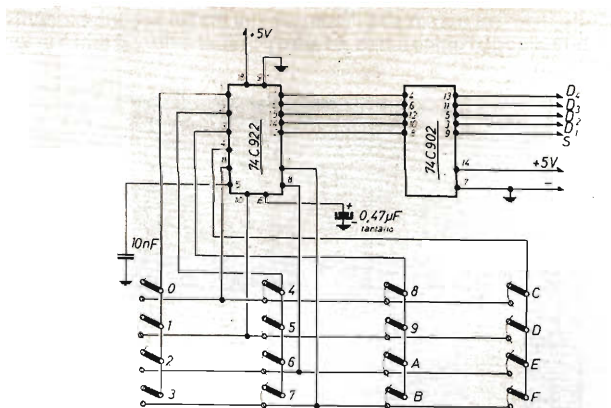
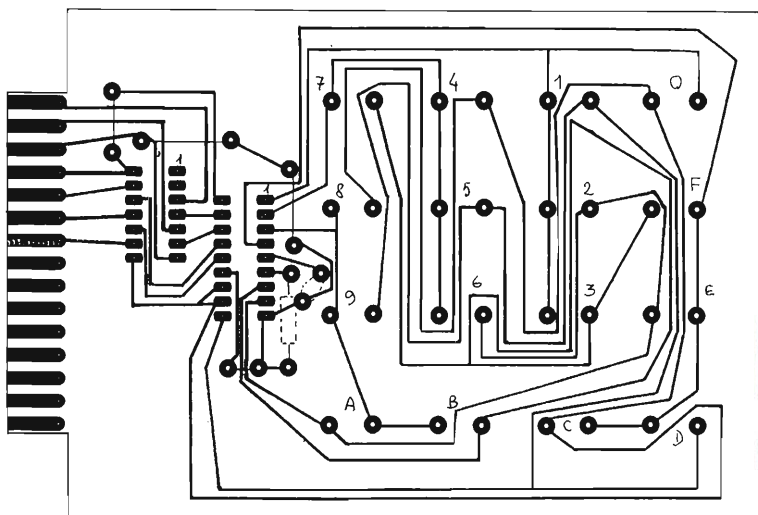
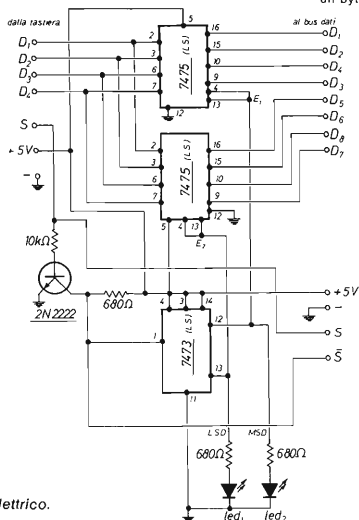
Distinguiamo: la tastiera e la distribuzione.
Iniziamo con la prima. Una matrice di 16 tasti (pulsanti) agisce su otto terminali di un integrato CMOS 74C922, costruito apposta come encoder (cioè codificatore) binario.
Traduzione: dette D4, D3, D1 le quattro linee di uscita (dati) e ponendo sui cappucci dei pulsanti le sedici cifre esadecimali elementari, si realizzerà la seguente tabella:

tasto premuto	D4	D3	D2	D1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Come vedete la solita tabella dei primi sedici numeri binari, cominciando da zero.
Segue una parte di un altro circuito integrato: esattamente 5 sezioni di un sestuplo buffer (separatore - amplificatore di corrente) non invertente 74C902.
Le diverse sezioni amplificano ognuna una linea d'uscita dell'encoderl Quattro linee e cinque sezioni?
Al diavolo la distrazione! avevo dimenticato di dirvi che esiste ancora una linea d'uscita: lo strobe.
Cos'è? beh, semplicemente un impulso che ha lo scopo di "abilitare" un dispositivo utente ad accogliere i dati presenti sulle altre quattro linee, quelle dei dati.

Dice: ma non si vede già che ci sono i dati, che bisogno c'è di un impulso che dica che ci sono, benedetti i progettisti, guarda tu che complicazioni ti vanno a tirare fuori.
Dice: inanzitutto lo strobe diventa attivo (nel nostro caso "alto", così come le altre linee) solo dopo che tutti i livelli dei dati sono diventati stabili. Ah, già, dimenticavo. Esiste un circuito "antirimbalzo" che elimina i falsi contatti -o meglio i loro effetti- quando si premono i tasti. È esso un'integratore, quindi ritarda l'insorgere del segnale d'uscita, che quindi non arriva immediatamente alle linee. In secondo luogo può venire usato -l'impulso di

strobe- come facciamo noi, cioè per comandare un flip-flop che "shifti" alternativamente a destra e a sinistra l'indirizzo del nibble sui due blocchi di linee dati! In terzo luogo l'unico limite ad altre possibili utilizzazioni è la fantasia di chi lavora.

Tastiera: "distributore".
Diagramma di temporizzazione.

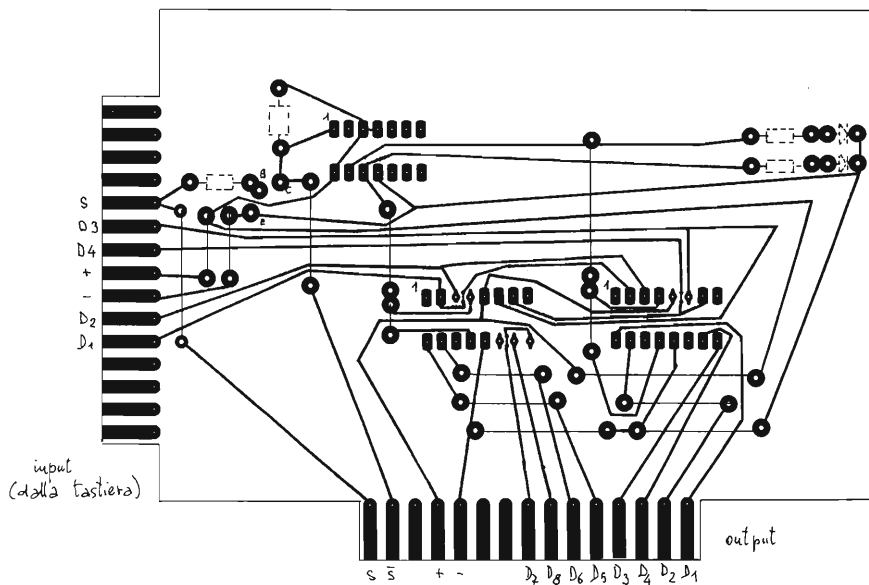
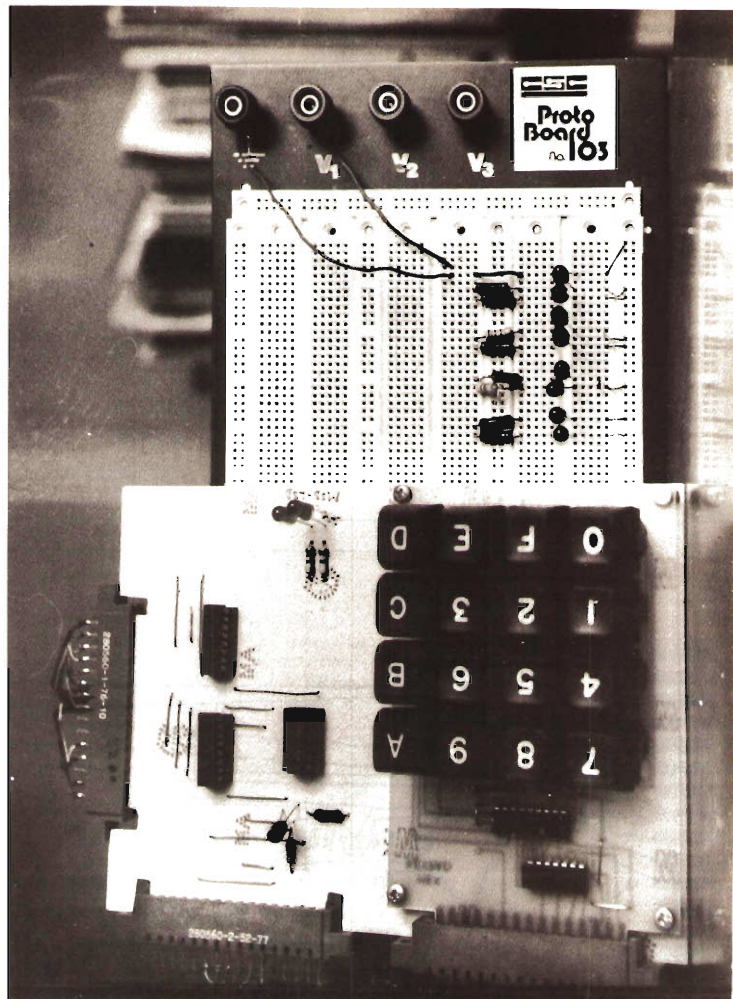
Ma avete visto, parlando, come si è complicato l'integrato che usiamo? Basta: ritorniamo a noi.
Abbiamo visto come funziona -si fa per dire- la tastierina.
Passiamo al distributore.

Dunque: dalla tastierina escono le quattro linee di dati e la linea di strobe: il tutto in logica positiva. Questo vuol dire che normalmente le cinque linee sono a potenziale zero, e che quando vengono portate a 5 V, ad esse è associato lo stato logico 1.
I D-Latch contenuti nei 7475 vengono usati per un temporaneo immagazzinamento dei livelli presenti al loro ingresso. Le informazioni presenti sotto tale forma ai terminali di entrata (dati) sono trasferite alle rispettive uscite quando il pin Enable (abilitazione) è alto, e le uscite seguono i livelli degli ingressi fino a che l'abilitazione rimane alta (a 5 V).

Keyboard: schema elettrico

Circuito stampato della keyboard, lato rame.

"Distributore": schema elettrico.

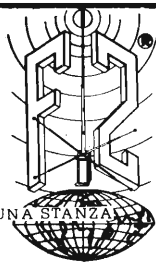
S
D3
D4
+
–
D2
D1

input (dalla tastiera)

output

S S̄ + – D7 D8 D6 D5 D3 D4 D2 D1

Circuito stampato del "distributore".
N.B. è visto dal lato rame. Le linee più sottili sono ponticelli (dal lato componenti).

Siccome il 7475 è un quadruplo D-Latch, e siccome l'abilitazione agisce su una coppia di tali flip-flop, ecco che ho bisogno di 2 pin di enable, uno per coppia: eccoli qui, i pin 4 e 13.
Appena i pin di enable vanno "bassi" (a tensione zero, livello logico Ø), le uscite rimangono invariate all'ultimo stato in cui sono state forzate, qualunque variazione abbiano i potenziali d'ingresso.

In questo modo io immagazzino, se i pin 4 e 13 del primo 7475 sono mantenuti alti, i dati del digit più significativo fino a che viene il momento di utilizzarli. Ora devo ottenere i dati del digit meno significativo: contestualmente all'istante in cui porto a zero il livello dell'enable del primo 7475, porto a 1 il livello dei pin 4 e 13 del 7475 n. 2.
Immagazzinare ora i dati di questo digit è un gioco da ragazzi.
Un led sui pin di enable mi indica quale 7475 sto caricando, quindi se sto lavorando sul digit più significativo o su quello meno significativo.
È abbastanza ovvio che posso agire, per sostituire i dati, o per correggere il digit che io avessi sbagliato di impostare, sulla parte di byte che è indicata dal led apposito.
Per ottenere la commutazione automatica del quartetto di D-Latch, uso un flip-flop del tipo JK, un 7473 usato come semplice binario forzando a livello alto i pin di J e K, ed entrando sul terminale di clock. Il 7473 è un doppio flip-flop. Io ne uso una sezione sola.

Per evitare consumi elevati, sarà bene usare integrati del tipo Low Schottky (LS), piuttosto che della serie normale, ma non è assolutamente tassativo. Dipende da quello che avete in casa.
Un'ultima cosa: per far si che la commutazione avvenga al termine della scrittura utile, cioè prima di scrivere il secondo digit, e siccome dispongo di uno strobe in logica positiva, mentre il clock del 7473 è sensibile alle transizioni verso l'alto, ho dovuto invertire il segnale di strobe. L'ho fatto con un transistore. E ho approfittato di tale fatto per deviare verso l'uscita un impulso di strobe di polarità opposta a quello fornito dalla tastierinaa: non si sa mai, potrebbe sempre servire, e, visto che c'è....
Volendo perfezionare il sistema per ricavare altri nibble su altre linee di dati, in parallelo, contemporaneamente a quelle già ora disponibili, potrete sostituire il 7473 con un 4017 contatore ad anello (di Johnson) decadico, collegando le uscite progressive ai vari enable di altri 7475 con tutte le entrate in parallelo, e le linee di uscita portate ai diversi connettori; l'entrata del contatore sarà collegata all'uscita (dello strobe negato), ed userò il terminale CPo, tenendo bassi MR e $\overline{CP1}$ (collegati a massa). Replicando il rimanente dello schema, potrò pilotare fino a dieci 7475.
Come farete a trovare dieci led di dieci colori diversi, è del tutto affar vostro. ✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳✳

per la grafica vettoriale

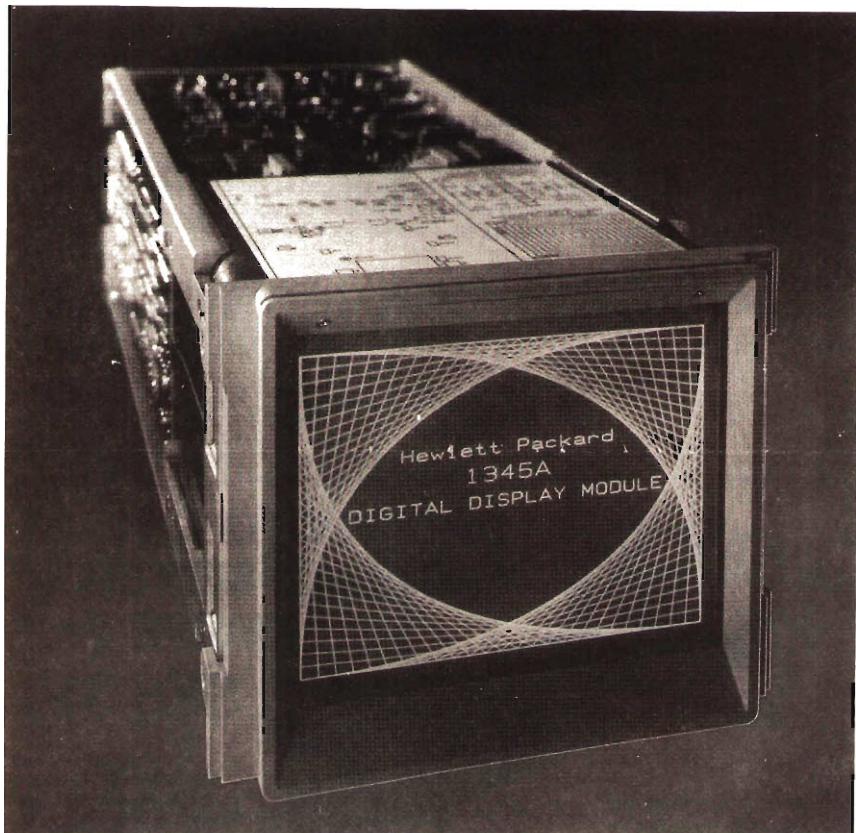
Modulo pilotabile
direttamente
dal
Data Bus del μp

Lear M. Casaroli

Un problema che abbiamo sia noi hobbisti che i tecnici di Laboratori, Docenti e studenti universitari, ingegneri o ragionieri d'Azienda è di disporre di un modulo di qualità per la grafica vettoriale, facilmente integrabile nella maggior parte di sistemi e/o strumenti di cui già disponiamo.

La HP ci aiuta a risolvere questo problema annunciando e rendendo disponibile sul mercato per la strumentazione degli OEM basata su μp, lo **HP 1345A,** un compatto schermo completamente autonomo per grafica vettoriale ad alta risoluzione. La HP informa, in più, che questo modulo costituisce il primo membro di una nuova famiglia di moduli display con interfonia digitale a elevate prestazioni.
Il nuovo HP 1345A produce grafici vettoriali o alfanumerici (o entrambi contemporaneamente) su uno schermo da 6", in risposta a comandi digitali inviati da un processore esterno.
Una interfaccia binaria permette la connessione al bus dei dati (Data Bus) 16 bit TTL dell'Utente.
Tipici strumenti del Tecnico o dell'Hobbista in cui può essere usato il modulo HP 1345A sono ad esempio gli analizzatori di spettro e di reti, gli analizzatori di funzioni, gli oscilloscopi con memoria digitale e gli analizzatori di Fournier, i tracciatori di curve, o comunque, qualunque apparecchio controllato da μp che richieda la rappresentazione di informazioni grafiche complesse e variabili con rapidità.

Grafica ad alta risoluzione

I 2.048 x 2.048 (4.194.304) punti indirizzabili sullo schermo permettono di ottenere una grafica veramente a elevata risoluzione, mentre la **qualità** dell'immagine viene ulteriormente migliorata dalla velocità di scrittura costante che assicura una luminosità uniforme.
Lo HP 1345A dispone anche, in opzione, di una memoria RAM da 4 kvettori per effettuare automaticamente il ripristino dell'immagine sullo schermo, lasciando la memoria dell'Utente e il μp disponibili per altri usi.

Una ulteriore flessibilità del modulo è data dalla possibilità di **programmare** tre diverse intensità luminose, l'accensione e lo spegnimento, quattro tipi di linee e quattro velocità di scrittura.
La funzione ΔX programmabile permette inoltre di **incrementare automaticamente l'ascissa** di una quantità decisa in precedenza dall'Utente, **dimezzando** la necessità di memoria e rendendo più rapida la visualizzazione.
I caratteri ASCII sono generati internamente in quattro diverse dimensioni **programmabili** e con quattro diverse direzioni per la didascalizzazione dei grafici, l'indicazione di tasti a funzione variabile, e altre applicazioni.
Il tempo medio di scrittura di un carattere è di ~ 15 μs, e permette di visualizzare centinaia di caratteri senza flicker.

Facile integrazione

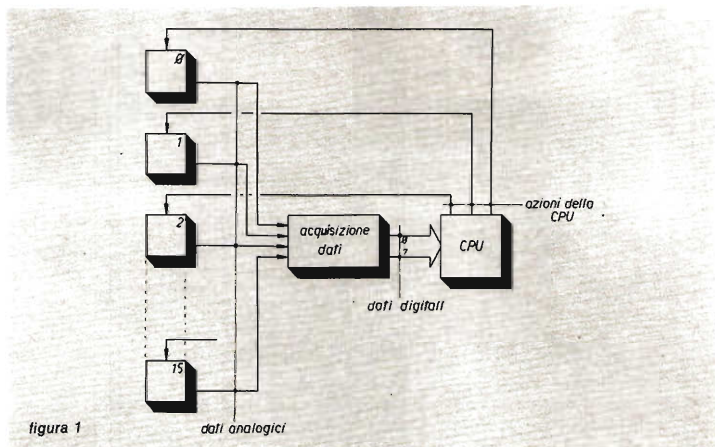
Lo HP 1345A è, come detto in epigrafe, facilmente integrabile nella maggior parte di strumenti o sistemi che richiedono un display di elevata qualità. L'integrazione è semplificata dall'interfaccia binaria a 16 bit che facilita il collegamento con µp a 8 o a 16 bit, dalla ROM di generazione dei caratteri e dal sistema interno di taratura con generazione delle sequenze di check-point e con Test Points facilmente individuabili.
Grazie alla tecnologia adottata per il circuito e al tubo a raggi catrodici a deflessione elettrostastica, il consumo totale del modulo alimentato a ± 15 V e a + 5 V, è inferiore a 25 W, con notevoli vantaggi anche dal puntò di vista della affidabilità. ✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲✲

Acquisizione dati da otto canali analogici per microcomputer

Antonio Anselmi

Il problema che qui si vuole risolvere consiste nel cablare un semplice sistema di acquisizione dati analogici, capace di fornire a una CPU otto diversi canali, contenenti informazioni binarie.

Prima di addentrarci nel vivo dell'argomento, facciamo alcune semplici chiarificazioni a riguardo delle specifiche richieste al sistema.
Innanzitutto **cosa è un sistema di acquisizione dati:** questo è uno strumento o circuito elettronico che scandisce automaticamente i dati prodotti da altri strumenti o elabora e registra le letture di tali dati al fine di una futura elaborazione.

figura 1

Da tale prima definizione si comprende chiaramente che un tipico sistema di acquisizione dati può "lavorare" in due diverse maniere, e cioè **on line** qualora il sistema acquisizione dati/CPU provveda alla elaborazione dei dati istante per istante, ovvero nello stesso istante in cui questi arrivano con la conseguente propagazione dalla CPU di segnali di controllo verso lo strumento o sensore dal quale i dati medesimi provengono. A tale scopo vediamo la figura 1.
Tipico è il caso in cui una CPU controlla il posizionamento di un pannello solare, al fine di mantenere quest'ultimo sempre in posizione ottimale (perpendicolare) nei confronti dell'irraggiamento proveniente dal sole, mediante la continua elaborazione di dati provenienti da sensori fotoelettrici posti sulla superfice del pannello (vedi la figura 2), metodo che va sotto il nome di "solar tracking"; oppure il caso simile per l'inseguimento di un satellite artificiale, manovrando l'antenna ricevente direttamente dalla CPU.

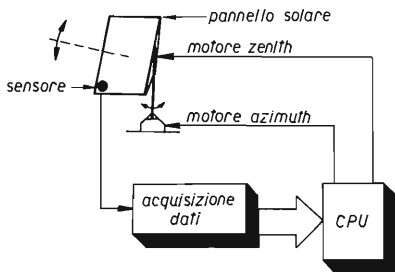



figura 2

Un'altra maniera di lavorare è **off line,** qualora il sistema acquisizione dati/CPU si limiti alla sola memorizzazione dei dati acquisiti per poi elaborarli in un secondo tempo (si noti, per i pignoli, che anche la "memorizzazione" può essere una forma di elaborazione).

Il sistema qui descritto, per altro semplicissimo (tre integrati e resistenze e condensatori vari), **si presta ad ambedue le funzioni indicate,** con una limitazione dipendente dal numero di bit disponibili sulle porte I/O della CPU in possesso. Tale problema sarà comunque affrontato nel prosieguo, e vedremo che sarà aggirabile abbastanza facilmente (leggi: compromessi).
Torniamo quindi sul discorso dell'acquisizione dati e sulle tematiche ad esso annesse e connesse.
Una volta decisa l'applicazione, on line oppure off line, dovremo stabilire il numero di strumenti e/o sensori esterni dai quali, in virtù della applicazio ne scelta, ci interessa prelevare quelle informazioni necessarie o alla elaborazione o alla memorizzazione: volendo si possono realizzare simultaneamente le due funzioni.
Il sistema qui descritto prevede in input un massimo di otto canali analogici: un numero sufficientemente alto per tutte le applicazioni sperimentali-hobbistiche ma che non impedisce anche ai più esigenti di sviluppare sofisticatissimi controllers. Ovviamente si possono ampliare tali condizioni ricorrendo a un maggior numero di canali in input: al solito una pesante limitazione viene dal numero di bit disponibili sulle porte I/O; questo perchè per indirizzare otto canali occorreranno tre bit ($2^3 = 8$),

mentre qualora si desideri un numero immediatamente maggiore di canali, occorreranno più bit per formare i relativi indirizzi (per completezza, occorreranno 4 bit per un numero compreso fra 1 e 16, 5 bit fino a 32, etc.).
Per tale motivo ho scelto un sistema a otto canali, totalmente flessibile in quanto ci consente di risparmiare un bit rispetto al sedici canali con la conseguente riutilizzazione per altri scopi di un bit.
Dobbiamo ora risolvere una terza specifica, ovvero quanti bit-dati sono richiesti per ogni canale. Mi spiego, abbiamo sinora visto che siamo in grado di acquisire fino a otto canali, contenente ciascuno una informazione analogica (solitamente una tensione proveniente dal voltmetro o da sensori quali fotoresistenze, NTC, etc.): come mi auguro che tutti sappiate, la CPU non è in grado di lavorare con tensioni continue ma solo con due valori binari e quindi è giocoforza operare una conversione da analogico a digitale (A/D conversion) su ogni canale che dovrà essere inputtato. E qui viene la scelta: è preferibile avere in uscita dal convertitore A/D una stringa di quattro bit? oppure 8 o 12? e in quale forma tali dati binari dovranno essere presentati alla CPU? in binario puro? in offset o in BCD? Inoltre, quale è il range ammissibile della tensione da applicare agli otto canali analogici? è una tensione sempre positiva oppure può scendere sotto lo zero?
A me pare ovvio che lo spazio disponibile non mi permette di illustrare un singolo ammenicolo per ogni combinazione delle precedenti, dirò subito, quindi, che il sistema in questione presenta le seguenti caratteristiche, in rapporto alla discussione cui sopra:

- numero di canali analogici da acquisire . . . . . da 1 a 8
- numero di bit-dati per canale. 8
- codice di rappresentazione alla CPU. . . . . . offset binary
- range della tensione input per canale. . . . . . da −5 a +5 V.

Una tale configurazione permette quindi di convertire in binario tensioni provenienti dai canali analogici a partire da −5 V (con stringa binaria corrispondente 0000 0000) fino a + 4,96 V (con stringa binaria corrispondente 1111 1111) con zero centrale (1000 0000).

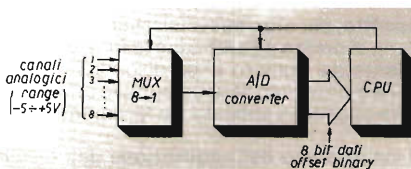



figura 3

Prima di entrare nella considerazione circuitale del sistema, è opportuno risolvere una ennesima specifica: ovvero in massimo numero dei canali acquisibili per secondo. Questo è un punto fondamentale per tutte le operazioni di acquisizioni dati. I dati possono, ad esempio, essere letti con lunghi intervalli di tempo (anche ore o giorni) mentre possono essere letti alla massima velocità consentita dal sistema.

Facciamo un esempio per ciò che riguarda questo ultimo caso.
Il nostro sistema ha una velocità di conversione pari a circa 1,8 ms, ovvero questo è l'intervallo di tempo richiesto per convertire un valore analogico di un canale in una stringa binaria a otto bit, per cui per la completa conver-

sione di otto canali occorrerà un tempo approssimativamente uguale a 14,4 ms (ho scritto approssimativamente in quanto dobbiamo tenere conto del tempo necessario al multiplexer per commutare i vari canali): ciò si traduce in circa 70 Hz per la conversione di tutti e otto i canali e in circa 556 Hz per un solo canale. A modificare, in senso negativo, tali tempi interviene il tempo necessario alla CPU per la lettura e la memorizzazione. Non credo comunque che la maggior parte di noi hobbisti abbia la necessità di acquisire dati che si evolgono a tali velocità: in pratica le esigenze si traducono in acquisire dati da semplici sensori con intervalli anche abbastanza lunghi (minuti o ore) tramite semplici routine di timer. Ad ogni modo, elenco di seguito le caratteristiche della parte riguardante la conversione A/D:

- risoluzione . . . . 8 bit in offset binary
- tempo di conversione . . . . 1,8 msec (massimo)
- non-linearità . . . . ± 1/ 2 LSB (massimo)
- drift di temperatura . . . . ± 50μV/ °C (massimo).

Iniziamo la discussione circuitale, elencando le linee di collegamento fra microcomputer e sistema di acquisizione dati:

linee di input
2 di controllo provenienti dal convertitore A/D
8 provenienti dal convertitore A/D costituenti il dato binario offset

linee di output
3 necessarie per indirizzare gli otto canali (massimo)
2 necessarie al controllo del convertitore A/D
1 necessaria al controllo del multiplexer

Fanno in totale 16 bit che dovremo collegare al microcomputer.

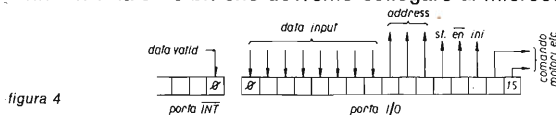



figura 4

Personalmente, disponendo di una CPU a 16 bit, ho adottato la configurazione di figura 4, ovvero: ● porta di sedici bit usata per l'input delle stringhe binarie (8 bit) ● controllo multiplexer (1 bit) + address (3 bit) ● controllo convertitore (2 bit) ● registro programmabile interruzioni usato per il controllo fine conversione (1 bit); mi rimangono quindi due bit liberi della porta per pilotare un digital power che azionerà due motorini. Ovviamente, si guadagna un bit come detto in precedenza, riducendo il numero dei canali analogici da acquisire.

Per ciò che riguarda le linee di controllo queste sono:

start conversion: impulso di inizio conversione che deve essere mandato al convertitore A/D. Tale impulso può essere mandato dalla CPU per intervalli di tempo prestabiliti oppure può essere fornito da un clock a parte (risparmiando in tale modo un altro bit sulla porta I/O) o un registro di timer interno al microcomputer.

data valid: proveniente dal convertitore A/D e che servirà per lanciare una richiesta di interruzione alla CPU. Tale linea indica, come vedremo dal diagramma temporale, che alle uscite del convertitore A/D sono presenti i validi valori binari della tensione presente al suo ingresso.

enable: necessaria per abilitare le uscite (3-state) del convertitore.
inhibit: necessaria per abilitare il funzionamento del multiplexer.

Le altre linee sono:

data input: otto linee contenenti il codice binario offset.
address: tre linee atte a formare l'indirizzo del canale che si vuole selezionare.

Ciò detto, veniamo all'analisi del semplice schema (vedi figura 5 e 6) del sistema presentato.

CD4051B, CD4052B, CD4053B Types

MAXIMUM RATINGS, *Absolute-Maximum Values:*

DC SUPPLY-VOLTAGE RANGE, (V_{DD}) Voltages referenced to V_{SS} or V_{EE}, whichever is more negative)	−0.5 to +20 V
INPUT VOLTAGE RANGE, ALL INPUTS	−0.5 to V_{DD} +0.5 V
DC INPUT CURRENT, ANY ONE INPUT	±10 mA
POWER DISSIPATION PER PACKAGE (P_D):	
For T_A = −40 to +60°C (PACKAGE TYPE E)	500 mW
For T_A = +60 to +85°C (PACKAGE TYPE E)	Derate Linearly at 12 mW/°C to 200 mW
For T_A = −55 to +100°C (PACKAGE TYPES D,F)	500 mW
For T_A = +100 to +125°C (PACKAGE TYPES D, F)	Derate Linearly at 12 mW/°C to 200 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR FOR T_A = FULL PACKAGE-TEMPERATURE RANGE (All Package Types)	100 mW
OPERATING-TEMPERATURE RANGE (T_A):	
PACKAGE TYPES D, F, H	−55 to +125°C
PACKAGE TYPE E	−40 to +85°C
STORAGE TEMPERATURE RANGE (T_{stg})	−65 to +150°C
LEAD TEMPERATURE (DURING SOLDERING): At distance 1/16 ± 1/32 inch (1.59 ± 0.79 mm) from case for 10 s max.	+265°C

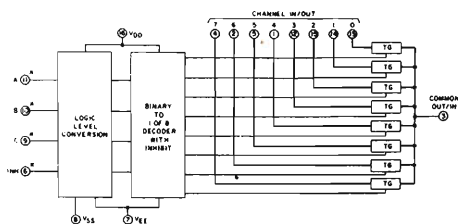
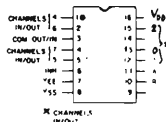
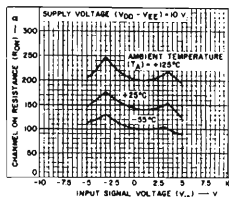
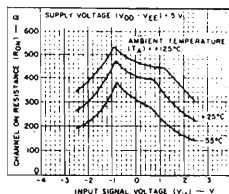



figura 5

figura 6

Il multiplexer usato è un COS/ MOS siglato 4051B, il quale multiplexa segnali analogici. Niente di particolarmente strano a proposito di tale integrato, se non il fatto del piedino di inhibit (active High) che abilita il multiplexer al funzionamento e la possibilità di programmare dall'esterno il range della tensione analogica agli otto ingressi. Infatti il segnale analogico

attraverso i TG, vedi figura 5, può variare fra V_{EE} fino a V_{DD}: quindi per ottenere la variazione ±5 V necessaria ai nostri scopi, dovremo operare i seguenti collegamenti:

V_{DD} collegato al +5 V;

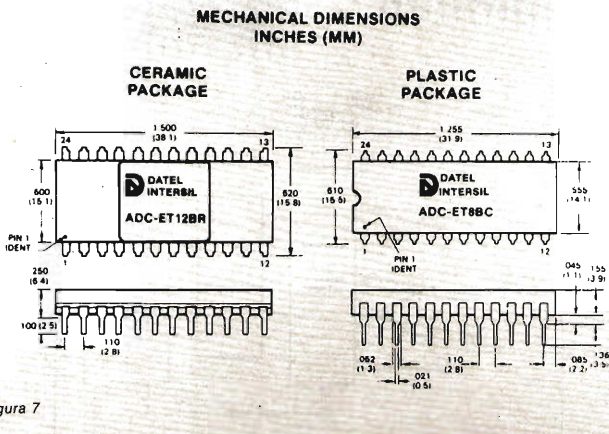
V_{EE} collegato al –5 V;

V_{SS} collegato allo zero centrale, massa del sistema.

Il canale selezionato con la cifra binaria agli ingressi A B e C del 4051 (piedini 11, 10 e 9) apparirà all'uscita (piedino 3) qualora il comando di inhibit sia disattivato (uno zero logico al piedino 6).

È enfatico dire che il cuore del sistema è costituito dall'integrato ET8BC (prodotto dalla Datel Intersil), la quale configurazione è visibile nelle figure 7 e 8.

Prima di descriverne il funzionamento, guardiamo insieme alcune caratteristiche.

figura 7

La prima cosa che salta all'occhio è la possibilità dell'integrato di fornire in uscita un numero variabile di bit: da 8, 10 e 12 bit. Come detto in precedenza, noi usiamo tale integrato per avere in uscita stringhe da 8 bit, quindi dovremo collegare le linee data input con i seguenti piedino dell'ET8BC: il piedino 12 rappresenta sempre il bit meno significativo, e quindi a tale piedino andrà collegata la linea meno significativa del data out.

Proseguiremo poi nei collegamenti, fino ad arrivare al piedino 5 che rappresenta il bit più significativo, nel caso di uscita a 8 bit. Ad ogni modo, ciò è esaurientemente illustrato nella figura 9.

Le uscite digitali dell'ET8BC possono pilotare due carichi del tipo Low Power TTL oppure un carico Low Power Schottky TTL, quindi sono compatibili con la tecnologia C/MOS. Quando l'input start convert è attivato da un impulso (largo almeno 500 ns!), la uscita busy va alta e inizia un ciclo, di circa 10 µs, durante il quale viene scaricata la capacità di integrazione nonché resettati i due contatori contenuti all'interno dell'ET8BC. La conversione vera e propria inizia dopo un impulso di reset interno. La fine conversione è segnalata da un impulso generato dal clock counter (sempre in-

terno all'integrato) oppure generato dal clock counter nel caso, spiacevole, di un overflow; tale impulso disabilita ulteriori ingressi verso i contatori e inizia un ciclo shutdown, della durata di circa 10 μs (a tale proposito si cerchi di consultare parallelamente il mio scritto, con il diagramma temporale di figura 10).

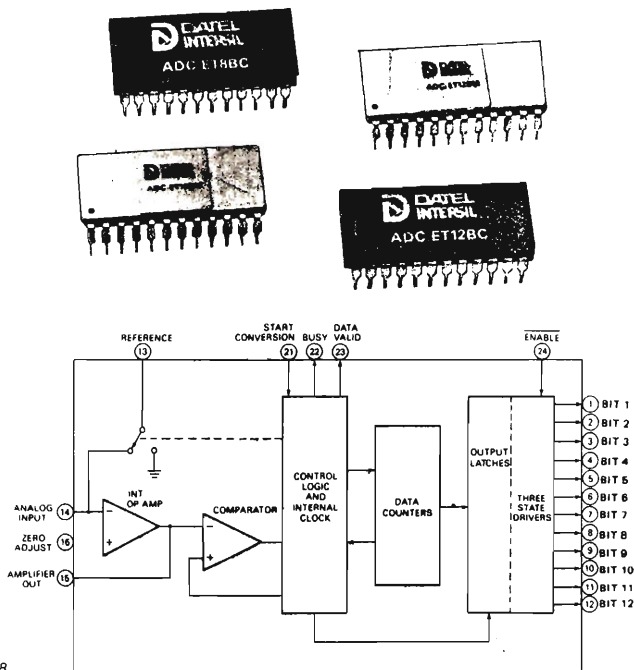



figura 8

INPUT/OUTPUT CONNECTIONS

FUNCTION	PIN	FUNCTION
BIT 1 (MSB 12 BITS)	13	REFERENCE
BIT 2	14	ANALOG INPUT
BIT 3 (MSB 10 BITS)	15	AMPLIFIER OUT
BIT 4	16	ZERO ADJUST
BIT 5 (MSB 8 BITS)	17	BIAS
BIT 6	18	−5V POWER
BIT 7	19	+5V POWER
BIT 8	20	GROUND
BIT 9	21	START CONVERT
BIT 10	22	BUSY OUTPUT
BIT 11	23	DATA VALID
BIT 12 (LSB ALL)	24	$\overline{\text{ENABLE}}$

figura 9

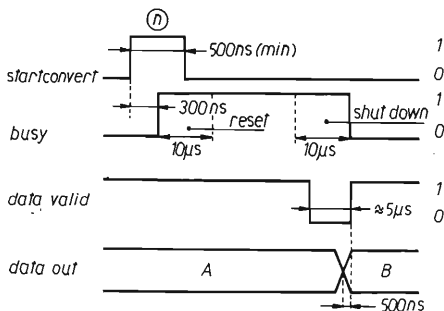



figura 10

A = dati relativi alla conversione iniziata all'impulso n−1.
B = dati relativi alla conversione iniziata all'impulso n.

All'interno dello shutdown, l'uscita data valid va bassa (segnale che useremo per lanciare l'interrupt) per circa 5 μs, al termine dei quali torna bassa l'uscita busy e sulle uscite digitali è presente il codice binario offset della conversione effettuata. A questo punto, dovremo inviare il segnale di enable all'ET8BC al fine di sbloccare le uscite e prelevare così l'informazione binaria.
Lo schema completo è visibile a figura 11: si noti la presenza dell'amplificatore operazionale 776 il quale serve a dare la corrente di offset dalla tensione di riferimento.

(i componenti a pagina seguente)

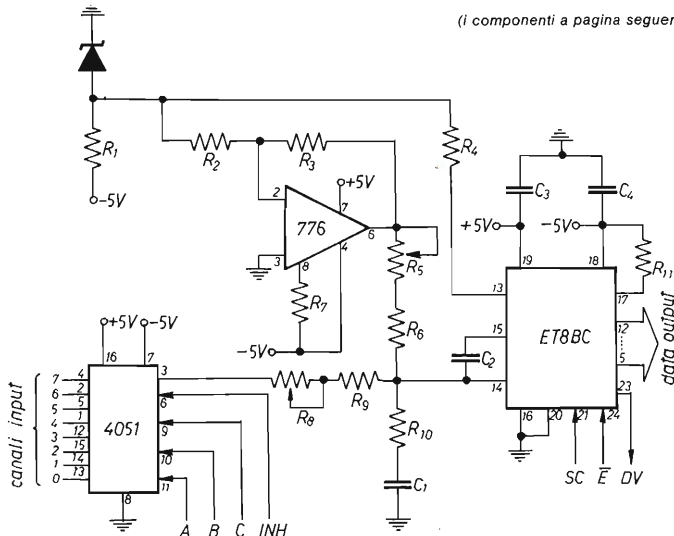



figura 11

Guardiamo attentamente lo schema, e cerchiamo di capire i vari valori assegnati ai componenti passivi.
Al piedino 17 dell'integrato (piedino di bias) va collegata una resistenza di polarizzazione del valore di 100 kΩ, con tolleranza massima del 10 %, un terminale di tale resistenza va collegato al negativo della alimentazione (—5 V); i piedini 18 e 19 vanno collegati, come vedesi in figura 9, rispettivamente al —5 e +5 V per la alimentazione. È consigliabile bypassare verso massa entrambi tali piedini, tramite due condensatori di valore 100 nF.
La tensione negativa di riferimento deve essere applicata al piedino 13, ed è ottenuta tramite uno zener e una resistenza da 3.600 Ω collegata al —5 V della alimentazione. La resistenza in serie all'ingresso della tensione di riferimento deve essere tale da far scorrere una corrente di 20 μA: usando la formula di Ohm avremo

$$R = V_{rif} / 20\ \mu A$$

ovvero, usando una tensione di riferimento di —1,22 V

$$R = 61\ k\Omega$$

la quale dovrà essere del tipo a film metallico, per ovviare agli inconvenienti degli sbalzi di temperatura. Logicamente, la precisione della conversione dipenderà in massima parte dalla precisione della tensione di riferimento, per cui si preferisce avere una regolazione del 4 % su tale tensione.
La resistenza in serie all'ingresso analogico è calcolabile dalla

$$R = FSR / 10\ \mu A$$

dove FSR è la variazione picco-picco della tensione in ingresso.
Il nostro sistema prevede una escursione da —5 a +5 V, per un valore picco-picco pari a 10 V; il valore di tale resistenza sarà quindi

$$R = 1\ M\Omega$$

Ad ogni caso, in serie a tale resistenza metteremo un trimmer multigiri, regolato in fase di taratura, del valore di circa 1 % di 1 MΩ, ovvero da 10 kΩ: tale trimmer ci servirà per la regolazione del guadagno.
Sull'uscita dell'amplificatore operazionale troviamo una serie di un trimmer (anch'esso multigiri) con una resistenza: tale servirà all'aggiustamento della corrente di offset. Gli altri condensatori sono ceramici.
Ad ogni modo questa è la lista dei componenti:

R_1 3.600 Ω
R_2 20 kΩ
R_3 20 kΩ
R_4 61 kΩ
R_5 2,2 kΩ, trimmer multigiri
R_6 244 kΩ
R_7 6,2 MΩ
R_8 10 kΩ, trimmer multigiri
R_9 1 MΩ
R_{10} 100 Ω
R_{11} 100 kΩ

C_1 270 pF
C_2 68 pF (capacità di integrazione)
C_3 100 nF
C_4 100 nF
(tutti ceramici)

zener LM 113

In figura 12 riporto la tabella della verità per decodificare gli address per gli otto canali analogici in ingresso al 4051.

input states		canale "ON"
inhibit	C B A	
0	0 0 0	0
0	0 0 1	1
0	0 1 0	2
0	0 1 1	3
0	1 0 0	4
0	1 0 1	5
0	1 1 0	6
0	1 1 1	7
1	X X X	nessuno

figura 12

A questo punto non mi sembra ci sia nulla da aggiugere a quanto detto, riporto solo nelle figure 13 e 14 le curve caratteristiche dell'integrato ET8BC.

Per ciò che riguarda il diagramma di flusso del software annesso al sistema di acquisizione dati, mi sembra che sia abbastanza estraibile: occorre solo molta attenzione nel conteggio dei tempi, quindi occorre fare bene i conti con il clock della CPU che voi usate e fare in modo da starci "larghi" (a tale proposito, ricordo inoltre di tenere bene in conto i ritardi di propagazione per l'ingresso inhibit del 4051 e dell'ingresso enable dell' ET8BC).

Ad ogni modo una tipica sequenza potrebbe essere questa:

- inizializzazione delle porte I/O;
- inizializzazione registri e puntatori;
- disabilitazione dell'interrupt;
- disabilitazione al 4051 (1 logico al piedino 6);
- A) formazione dell'indirizzo del canale scelto;
- B) out dell'indirizzo verso il 4051;
- abilitazione al 4051;
- start conversione verso ET8BC;
- abilitazione dell'interrupt;
- accettazione dell'interrupt;
- abilitazione all'uscita digitale dell'ET8BC (zero logico al piedino 24);
- disabilitazione al 4051;
- acquisizione in memoria del dato letto sulla porta input;
- disabilitazione dell'interrupt.

A questo punto la CPU entra in un loop di ritardo che può essere vantaggiosamente utilizzato o sistemando i dati letti in memoria e incrementando il relativo puntatore o elaborando i dati finora letti; quando la CPU rientra dal loop di ritardo si procederà per due strade diverse, a seconda che si vogliano dati da un solo canale o da n canali:

tornare al punto A	incremento dell'indirizzo tornare al punto B

procedendo naturalmente a una verifica, che consenta alla CPU di sapere esattamente su quale canale agisce il multiplexer (ad esempio, se è il caso di tornare sul primo canale scelto in quanto sono gia stati "esplorati" tutti i canali scelti per l'acquisizione).

TYPICAL PERFORMANCE CURVES

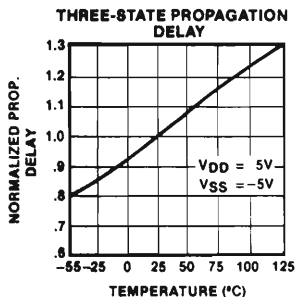
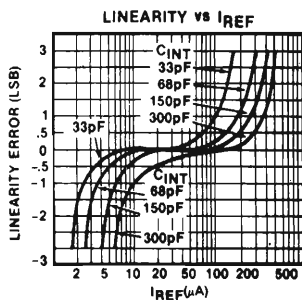
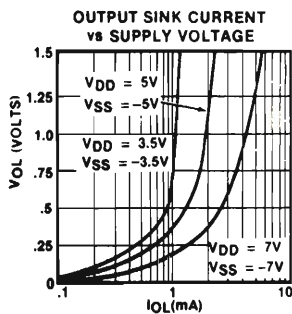
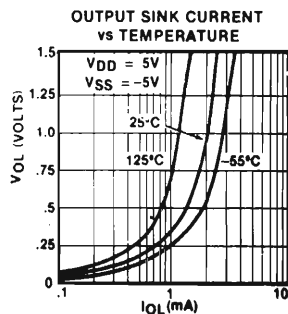
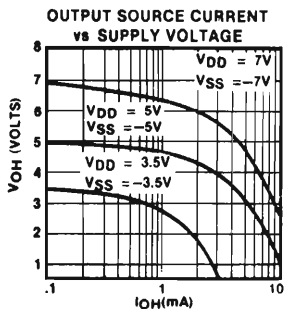
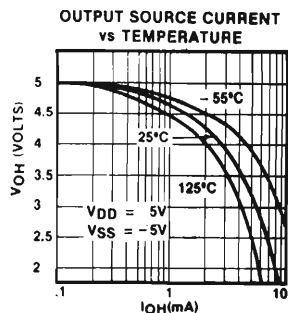



figura 13

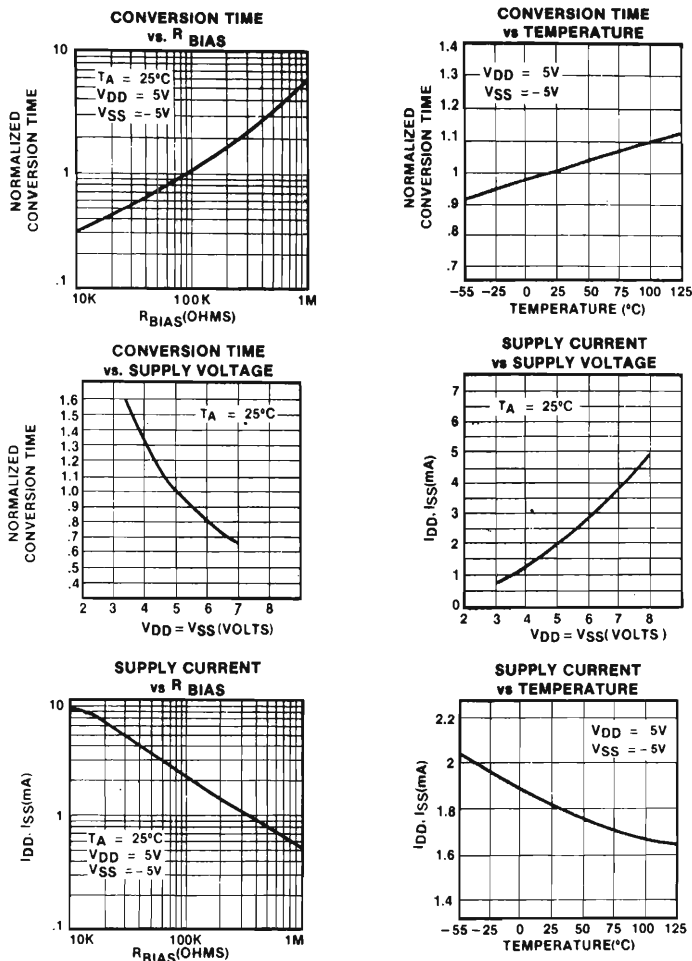



figura 14

Naturalmente ciascuno potrà giocare quanto vuole con le istruzioni possibili, fissare a un livello (0 o 1) uno degli indirizzi o addirittura tutti e tre gli indirizzi, nel caso in cui si voglia acquisire dati da un solo canale; in tale modo si risparmieranno tre bit e un bel po' di istruzioni.
Non aggiungo altro se non un buon lavoro e buon divertimento. ✻✻✻✻

come una intuizione può trasformarsi in un incubo

La prova del nove

ing. Luca Crispa

una macchinetta che vi dice se i vostri calcoli sono esatti

Era cominciato tutto come un gioco.
Le vacanze estive stavano per terminare, e ci si trovava t,utti immersi nel mondo della scuola, chi coinvoltovi direttamente —come i figli, e il sottoscritto— chi indirettamente, ma non meno intensamente, come la —rispettivamente— moglie e madre.
Riincomincia la scuola, dunque, e con essi i relativi problemi. Compresi quelli di aritmetica.
Che toccava a me, come padre e "testa d'uovo" della famiglia, sbrogliare.

La soluzione che la tecnologia ci mette davanti: l'eterna macchinetta calcolatrice elettronica, non soddisfaceva il mio istinto di educatore.
Avrebbe soddisfatto, ben s'intende, quelle di scansafatiche delle figlie. La maggiore delle quali, al mio ennesimo rifiuto di permetterle l'uso della TI-58, con l'aria di darmi l'ultima mazzata, dimostrando la sua ragionevolezza contrapposta alle mie dittattoriali pretese, si diede invece la classica zappa sui piedi.

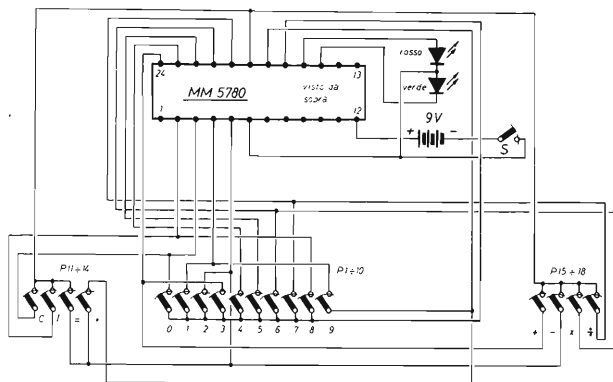
"Non è che io non sappia impostare le soluzioni, nè risolvere i problemi — cinguettò la sciagurata— *che il profe mi assegna. Nè (che Dio ci guardi) mi spaventano le operazioni pur nella loro immane complessità.*
Quello che mi sgomenta —continuò angelica— *è il non sapere se il risultato delle varie operazioni è esatto. Ora, se tu mi facessi una trappoletta, un aggeggetto, una specie di "prova del nove" elettronico, che mi eliminasse la necessità di perderci tempo, io non avrei bisogno della calcolatrice* (ed era certa che fosse impossibile, aveva già girato tutte le rivendite di materiale elettronico della zona) *che, credimi, non mi serve ad altro che a controllare il risultato del mio lavoro.*
Se invece, non puoi farlo —e lanciò l'ultimo perfido siluro, unendo in un solo mazzo la mia incompetenza e l'alta autorità materna—, *anche la mamma è d'accordo che io possa usare la tua macchinetta* (la TI-58, capite gente, stampante inclusa ...)".

Notti insonni.

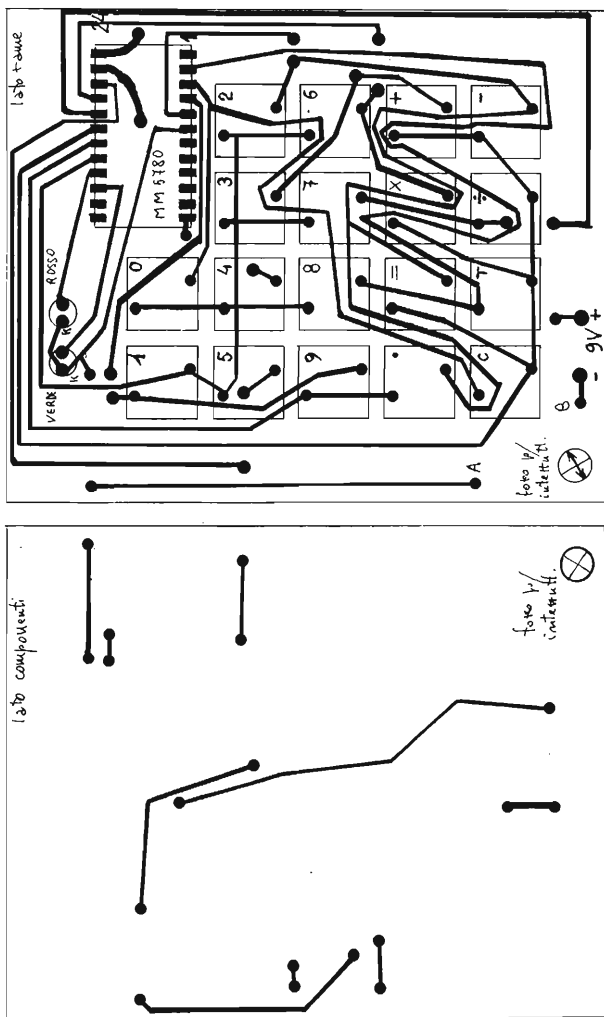
E riprendi a sfogliare i tuoi libri.
Mentre lei, la figlia, radiosa, ti interdice l'uso della TI-58 e ti esaurisce le scorte di carta termica.

Evviva la National!
Il grido che lanciavo quel giorno riecheggiava ancora per la casa, e mia figlia —tutta ingrugnata— si è vista riespropriata della calcolatrice di suo padre e, dal giorno di Natale, lavora con l'aggeggino che vi presento e che ho ribattezzato, in ricordo dell'ukase, **"prova del nove".**

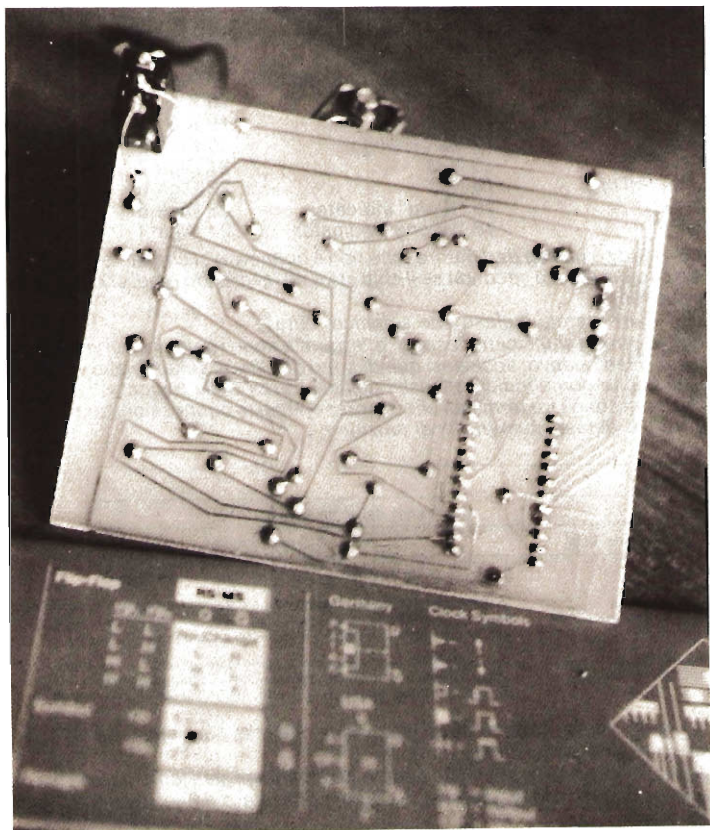
Il tasto magico

Il mio —si fa per dire, ché di mio c'è solo il circuito stampato— progettino si sviluppa attorno al circuito integrato MOS-LSI **MM5780** della **National.**

La "prova del nove": diagramma elettrico.

È questo un componente che utilizza un processo di produzione a gate metallico (è questo il significato della M di MOS), canale P (ovviamente di silicio, il semiconduttore che dà la S alla parola MOS, mentre la O rappresenta lo strato di biossido di silicio tra M e S), e tecnologia che combina i due modi "svuotamento e riempimento" ovvero "enhancement e depletion".
L'obiettivo principale della sua produzione è stato uno sviluppo a basso costo per permettergli di assolvere il compito di "giocattolo educativo", anche esso a basso costo.
Infatti, oltre al circuito integrale stesso, l'intero set richiede 18 tastini, un interruttore, uno zoccolo a 24 pin, un circuito stampato, due led a colori diversi, un connettore per pila, e una pila a 9 V.
Tutto il rimanente del circuito, compreso l'encoder da tastiera, i circuiti antirimbalzo, la generazione di impulsi di clock e la temporizzazione, i buffer verso i led, sono tutti "on-chip", cioè inclusi nel componente e non richiedono alcun componente esterno.
I problemi vengono fatti entrare nella macchina in forma algebrica, esattamente come essi vengono scritti dall'allievo nelle sue indicazioni, sulla pagina del quaderno.
Lo studente provvede la risposta e il fattore mancante e, al termine di tale operazione, preme il tasto "TEST". Si accenderà un led verde se la risposta e il dato inseriti erano esatti, altrimenti il led che si accenderà sarà rosso. Si

può ritentare la soluzione del problema, e —cosa da non trascurare e che la rende adatto per allievi di scuole anche di grado superiore all'elementare— si possono trattare anche numeri negativi. Ma questo risulta più chiaro degli altri esempi a corredo, in calce all'articolo.

Layout del circuito stampato, in grandezza vera, per ambedue le facce.

Avrete gia capito che il bello di questa macchina è che essa NON fornisce il risultato di nessuna operazione, ma si limita a segnalare se quello ottenuto è esatto: quindi nemmeno il professore di matematica più intransigente può obiettare nulla al suo uso.

La vita della batteria è stimata da 10 a 30 ore di lavoro effettivo, quindi lunghissima, a seconda del suo stato.
Nel chip esiste anche un circuito comparatore di tensione: il suo scopo? Quando la tensione della batteria scende sotto il livello operativo, esso disabilita ambedue gli indicatori a led: nessuno di essi si accenderà premendo il tasto "TEST".

Un led inserito con il suo resistore adeguato tra il pin 1 e quello 12 indica se il chip è "ready", ovvero se sta svolgendo un'operazione: a me non importava, quindi non l'ho inserito.
Esistono altre prestazioni valide dal punto di vista tecnico, ma per lo stesso motivo non ne tratto, invitando gli interessati a leggersi il manuale MOS/ 1980 della National, reperibile presso tutti i fornitori di letteratura tecnica.

Le righe che precedono, unitamente allo schema elettrico e al layout del circuito stampato potrebbero gia bastare per una buona parte dei miei lettori, ma per quelli che desiderano uno sguardo maggiormente approfondito, ecco qui alcuni altri:

appunti tecnici

Come era già evidente dal fatto che si utilizza uno zoccolo dual-in-line da 24 pin, anche il nostro circuito integrato presenta lo stesso numero di piedini. Le loro funzioni sono:

1 Ready	*9 N C (non connesso)*	*17 Key input 1 (K 1)*
2 Digit 9	*10 N C*	*18 K 2*
3 Digit 1	*11 N C*	*19 K 3*
4 Digit 2	*12 V_{ss} (+ alim.)*	*20 Digit 8*
5 Digit 3	*13 N C*	*21 Digit 7*
6 V_{dd} (-alim.)	*14 N C*	*22 Digit 6*
7 N C	*15 "Right" indicator*	*23 Digit 5*
8 N C	*16 "Wrong" indicator*	*24 Digit 4*

N.B. Ready è un flag (indicatore) che non è connesso al circuito esterno, ma che cambia stato a seconda se il chip sta o no lavorando (vedi anche il testo precedente).
Digit è una parola inglese che significa "cifra".
Right e Wrong significano "esatto" e "errato" e sono i piedini a cui viene connesso il led relativo; anch'essi quindi sono dei "flag", soltanto che vengono usati anche come indicatori visivi dell'esattezza della risposta.
Key input sono i tre terminali che rappresentano le connessioni comuni per le tre file di tasti, numerici e di funzione.

tabella 1

Descrizione del segnale "ready"	
Funzione del chip	*Segnale ready*
In riposo	*Il flag ready sta a riposo, al potenziale logico High (V_{ss})*
Tasti premuti e svolgimento operazioni	*Quando viene premuto un tasto, viene dato il via alla temporizzazione che assicura stabilità sugli eventuali rimbalzi. Ready rimane High fino a che tale tempo è terminato, e il tasto è premuto: ritorna poi Low (V_{dd}).*
Tasti rilasciati e ritorno allo stato di riposo	*Ready rimane Low (basso) fino a che il tasto rilasciato ha terminato il rimbalzo e il chip è tornato allo stato di riposo. La transizione da basso ad alto segnala il ritorno allo stato di riposo, in cui il livello logico rimane alto.*

La tabella sopra riportata dà un'idea molto chiara di cosa succede ed è rilevabile oscillograficamente al pin 1.
Quella invece che segue indica come cambiano gli stati logici ai pin 15 e 16 –rilevabili con i led indicatori–, e come interpretare quindi la luminosità di tali led.

tabella 2

Tavola della verità degli indicatori		
Condizioni del chip	*Uscita degli indicatori*	
	pin 15	*pin 16*
L'ultimo tasto premuto è stato quello di "TEST", ed è stata data una risposta corretta.	*High*	*Low*
L'ultimo tasto premuto è ancora quello "TEST", ma la risposta fornita è sbagliata, oppure il problema è scorrettamente impostato, o ancora esiste una condizione di "overflow" (superamento della capacità numerica del chip).	*Low*	*High*
L'ultimo tasto premuto è diverso da quello "TEST", ma non si verificano altre condizioni di errore.	*Low*	*Low*
L'ultimo tasto premuto è "CLEAR" (cancellazione).	*Low*	*Low*
La tensione di batteria è scesa sotto il valore di funzionamento del 5780. Si verifica premendo il tasto "TEST". Se non si accende alcun led, la batteria deve essere sostituita.	*Low*	*Low*

N.B. gli errori, o l'impostazione scorretta a cui si fa cenno di sopra, sono esclusivamente di tipo matematico (il chip non è in grado di dirvi se la formula che state usando è esatta o no!), come ad esempio una divisione per Ø (zero)!

Passiamo ora alle:

operazioni con i tasti ed esempi di problemi

Tasto "CLEAR": *cancella tutti i registri e riporta la macchinetta allo stato preferito: quello di riposo.*

Entrata dei numeri: *il primo entrato pulisce il registro di ingresso e fa "entrare" il numero nel registro della cifra meno significativa (LSD = Least Significant Digit), spegnendo contemporaneamente l'eventuale led acceso.*
I prossimi numeri fino all'ottavo "shiftano" (spostano) quelli precedenti uno alla volta, sospingendoli nel registro adiacente, fino a quello MSD (cifra più significativa, Most Significant Digit). Il nono e seguenti vengono semplicemente ignorati, e non causano condizioni di errore. Se esiste un punto decimale, i digit accettati sono solo 7, e gli altri vengono ignorati, con le stesse condizioni.

Punto decimale: *per noi corrisponde a "virgola decimale". Viene premuta per inserire tale segno nella cifra che si sta impostando.*

+ − x ÷ *(somma, sottrazione, moltiplicazione, divisione). Premendo uno di tali tasti dopo aver inserito un numero, si obbliga l'entrata a riconoscere i numeri successivi come parte del secondo; predispone il calcolatore ad effettuare l'operazione indicata, e registra la funzione premuta come la prossima operazione da effettuare dopo aver fatto entrare il numero seguente. Pressioni successive di diversi tasti di funzione, senza che vengano inseriti numeri, o punto decimale,*

cancellano le pressioni precedenti (cioè, se io premo, ad esempio, + — + x l'operazione che verrà svolta, se io tra i segni di operazione non ho inserito numeri, sarà quella di moltiplicazione.

= (eguale): la prima pressione dopo aver fatto entrare un numero provoca il riconoscimento del termine di tale numero da parte del chip, e fa effettuare la operazione precedentemente registrata, provoca la registrazione della pressione del tasto "=". Se tale pressione avviene dopo un segno di operazione, senza interpolazione di numeri, essa viene semplicemente ignorata, a meno che il segno di operazione non sia "x", cioè un prodotto. In tal caso il chip effettua un'elevazione al quadrato.

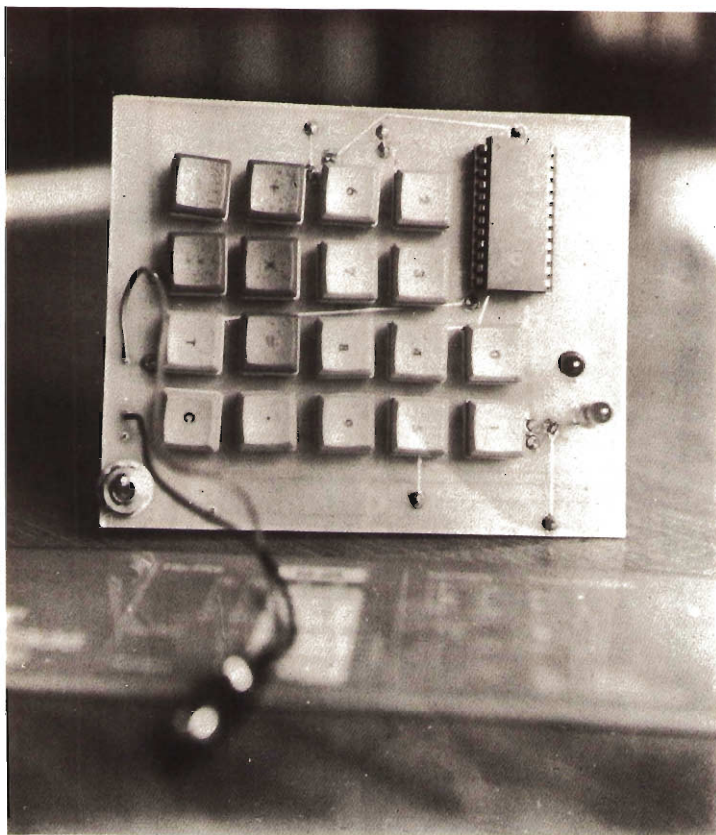
Entrata dei risultati: i risultati vengono fatti entrare semplicemente immettendoli premendo i tasti "digit" dopo l'"eguale", e "battendo" il numero che si prevede sia il risultato. Esso viene inteso come positivo (e NON SI DEVE battere il "+" prima di esso), mentre SI DEVE battere il "–" prima di esso se si prevede un risultato negativo.

TEST: il tasto T viene usato per terminare il calcolo e per effettuare il confronto tra il risultato fornito dallo studente e quello che il calcolatore ha effettuato e immagazzinato. Se essi coincidono si accende il led verde, altrimenti quello rosso. In tale ultimo caso NON È SUFFICIENTE battere un altro "risultato", ma bisogna reimpostare l'intera operazione.

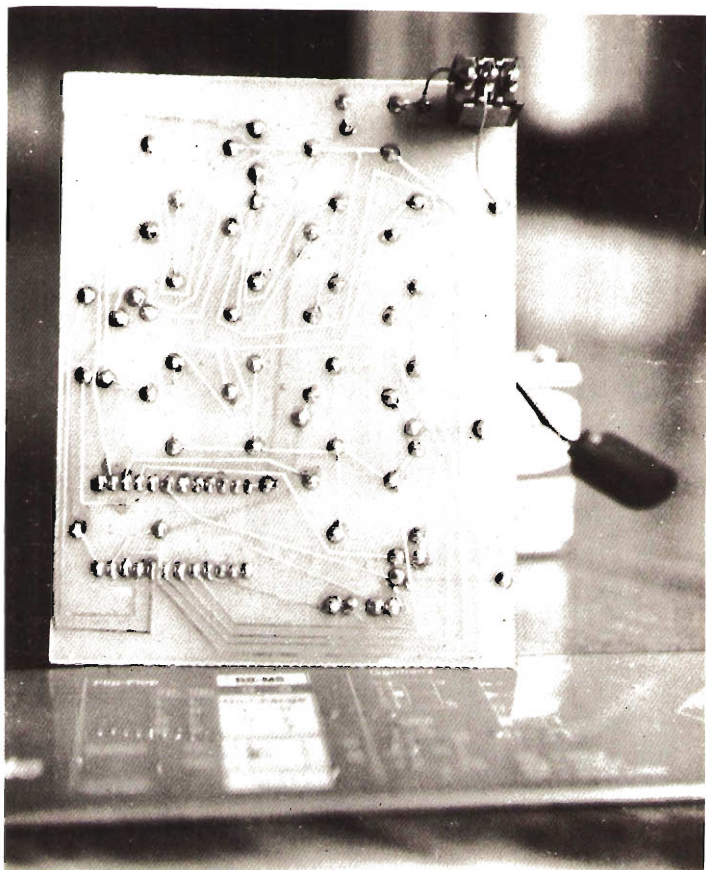
Infine, come risulta da quanto detto, il nostro aiutante è in grado di effettuare operazioni "in serie", cioè nell'ordine in cui vengono battute, senza riconoscere alcuna gerarchia, nè poter usare le parentesi per imporla: attenzione, quindi!

Esempio di somma:

voglio effettuare l'operazione 7 + 18 = ... e verificare il mio risultato

tasto premuto	led acceso	commenti
C		
C	nessuno	il doppio "CLEAR" è necessario all'accensione dell'apparecchio
7	nessuno	
+	nessuno	
1	nessuno	
8	nessuno	
=	nessuno	
2	nessuno	
5	nessuno	
TEST	verde	risposta esatta
Se invece, dopo l'eguale, io avessi avuto		
2	nessuno	
4	nessuno	
TEST	rosso	risposta errata

Per poter effettuare la verifica, avrei dovuto ripetere tutta la routine gia effettuata, iniziando dalla prima riga (però se non l'ho spento, basterà un solo C..., o addirittura nessuno, visto che il TEST ha gia effettuato un CLEAR) e fornendo una nuova risposta.

Altro esempio: trovare y in 9 + y = 17

tasto premuto	*led acceso*	
9	*nessuno*	*l'eventuale indicatore acceso di spegne.*
+	*nessuno*	
8	*nessuno*	*introduco la mia ipotesi per y*
=	*nessuno*	
1	*nessuno*	
7	*nessuno*	*termina la mia risposta*
TEST	*verde*	*OK*

Ancora un esempio: $8^2 = ...$

8	*nessuno*	*l'eventuale indicatore si spegne*
x	*nessuno*	
=	*nessuno*	*il complesso x = effettua l'elevamento alla seconda potenza*
6	*nessuno*	
4	*nessuno*	*termino la risposta*
TEST	*verde*	*OK*

L'ultimo:

7	*nessuno*	*l'eventuale led acceso si spegne*
+	*nessuno*	
8	*nessuno*	
+	*nessuno*	
1	*nessuno*	
x	*nessuno*	
3	*nessuno*	
÷	*nessuno*	
1	*nessuno*	
2	*nessuno*	
=	*nessuno*	
4	*nessuno*	*Introduco la risposta*
TEST	*verde*	*OK: ho effettuato* $\frac{(7+8+1)x3}{12} = 4$

Infine: cercavo l'altro giorno la TI 58.
L'aveva mia figlia: si stava stampando i "bioritmi": vuoi vedere che dovrò cercare un integrato LSI che li calcoli?
Incubo! Ma per fortuna c'è la National! *********************

COMMUNICATIONS COMPUTER TONO 9000 E

Tono Θ - 9000 E sistema per comunicare in CW/F1 (Murray/Ascii) governato mediante µP.

NUOVO TONO THETA

Versione radicalmente nuova dell'ormai nota Θ 7000 E con flessibilità operative ancora piú estese.
Questo modello infatti conserva le fondamentali possibilità operative precedenti: CW - RTTY, estendendo nel contempo l'apparato all'uso di video grafici nonché alla redazione di testi.
È possibile perciò scrivere ad esempio una lettera, quindi mantenendola in memoria, apportarvi tutte le correzioni successive del caso, ed infine effettuare la stampa del testo corretto in modo impeccabile mediante l'annessa stampante opzionale HC-900. La memoria è capace di 14.000 caratteri, mentre ciascuna "pagina" visualizzata contiene 24 linee di 80 caratteri ciascuna. Mediante l'apposita sonda (o penna luminosa) è possibile tracciare sullo schermo grafici o disegni che potranno essere quindi trasmessi senza dover procedere al tradizionale sistema dattilografico adottato sinora per la trasmissione di disegnini marginalmente abbozzati.
La memoria elastica dispone di 3120 caratteri, per cui si potrà preparare la risposta per il corrispondente - mediante la possibilità offerta dallo schermo dimezzato - nel leggere il messaggio in arrivo, quindi procèdere all'emissione con la massima velocità consentita.

Tutte le funzioni indispensabili al traffico RTTY sono state automatizzate, perciò i sani principi operativi acquisiti con le versioni meccaniche (ad es. Carr. Ret. - Line Feed, ecc.) sono divenute ormai una programmazione normalizzata. Lo speciale demodulatore CW si adatta bene a diverse velocità di manipolazione, mentre, durante la trasmissione oltre che alla velocità, è possibile modificare pure la "pesatura". Le funzioni RTTY includono le lettere e frasi standard per la regolazione e la sintonia iniziale necessaria per il traffico quali ad es.: RY-RY; The Quick Brown Fox.....
Speciale attenzione inoltre è stata devoluta all'apprendimento della telegrafia: un generatore ad accesso casuale emette - a comando - i piú disparati caratteri in CW, per cui l'operatore inesperto, regolando velocità e pesatura, potrà affidarsi a questo modo semplice, ma sempre sicuro di comunicare.

MARCUCCI S.p.A.
Exclusive Agent
Milano - Via F.lli Bronzetti, 37 (ang. C.so XXII Marzo) - Tel. 7386051

TONO

BES Milano

a partire da lire
680.000